リモート・コントロール・マニュアル

DS-4372/4372L

DS-4374/4374M/4374L

DS-4354/

DS-4354M/4354ML

DS-4262/4262ML

DS-4264/4264M/4264ML

KML040141

A701-422500(B)

トレードマーク

全ての商品，名称，サービスは、各社の商標または登録商標です。以下のものを含みます。
Ethernet は、Xerox Corporation の登録商標です。
Microsoft，MS-DOS，Windows は、Micorsoft，Inc の登録商標です。
NetBIOS は、International Business Machines Corporation の商標です。
ActiveX，Visual Basic，Excel の登録商標です。
Java は、米国およびその他の国における米国 Sun Microsystems，Inc の商標または登録商標です。

履　歴

◇2001 年 1 月　第 1 版発行
◇2001 年 3 月　第 2 版発行
◇2001 年 7 月　第 3 版発行
◇2002 年 11 月　第 4 版発行

目　次

第１章　はじめに

サービス体制 1-1
出荷品の点検 1-1
保証 1-1
修理 1-1
修理品の返却 1-1
保証延長 1-1
ドキュメント 1-1
技術サポート 1-2

第２章　リモート・コントロールについて

GPIB 実装標準 2-1
プログラム・メッセージ 2-1
コマンドとクエリ 2-2
ローカル・ステートとリモート・ステート 2-2
プログラム・メッセージ・フォーム 2-3
コマンド / クエリ・フォーム 2-4
応答メッセージ形式 2-7

第３章　GPIB操作

GPIB 構造 3-1
インターフェイス機能 3-1
アドレス指定 3-2
GPIB 信号 3-2
IEEE 488.1 標準メッセージ 3-3
GPIB 転送のプログラミング 3-5
サービスリクエストのプログラミング 3-9
オシロスコープ・ポール (Instrument poll) 3-10
ハードコピー・デバイスの駆動 3-15

第４章　RS-232-C操作

RS-232-C ピン配列 4-1
RS-232-C 構成 4-2
GPIB コマンドをシミュレートするコマンド 4-5

目　次

第５章　波形構造

波形の論理データ・ブロック 5-1
INSPECT?クエリ 5-2
WAVEFORM?クエリ 5-4
WAVEFORM コマンド 5-9
波形クエリをさらに制御 5-9
高速波形転送 5-10

第６章　ステータス・レジスタ

ステータスとサービスリクエストレポートの概要 6-1
ステータス・バイト・レジスタ（STB) 6-3
標準イベント・ステータス・レジスタ（ESR) 6-4
標準イベント・ステータス・イネーブル・レジスタ（ESE) 6-5
サービスリクエストイネーブル・レジスタ（SRE) 6-5
パラレルポール・イネーブル・レジスタ（PRE) 6-5
内部ステート変更ステータス・レジスタ（INR) 6-6
内部ステート変更イネーブル・レジスタ（INE) 6-6
コマンド・エラー・ステータス・レジスタ（CMR) 6-6
デバイス依存エラー・ステータス・レジスタ（DDR) 6-6
実行エラー・ステータス・レジスタ（EXR) 6-7
ユーザ・リクエスト・ステータス・レジスタ（URR) 6-7

システム・コマンド ： 特別セクション ..... SC-1 to 204

付録 A ： GPIB プログラム例
付録 B ： 波形テンプレート

イーサネット・オプション

概要 E-1-1
PC またはネットワークへの接続 E-2-1
オシロスコープ・ソフトウェア・ツール E-3-1

サービス体制

出荷品の点検

この度は、弊社製品をお買い上げいただき有り難うございました。出荷時に入念な点検を行い発送致しておりますが、製品がお客様のお手元に到着いたしましたら、ご面倒ですがすみやかに内容物の点検をお願いいたします。同封のリストと照らし合わせて過不足がないことをお確かめください。万一、過不足または、運送上で損傷等がございましたら、岩通計測株式会社か岩通ティー・エム・イー・サービス株式会社または担当の代理店までご連絡ください。ご連絡が遅れますと、対応できなくなる場合がございますのでご注意ください。

製品保証

この製品は、お客様に安心してお使い頂くために下記の保証をいたします。

◆ 保 証 期 間　　ご納入後１年間保証いたします。

◆ 保 証 条 件　　万一、保証期間内に当社の責任による不測の故障などが生じた場合には無償修復いたします。
連絡先は、サービスネットワークをご利用ください。

修　理

修理の必要が生じた製品は、岩通計測株式会社または岩通ティー・エム・イー・サービス株式会社にご返送下さい。保証期間内の製品に関しては無償で修理いたします。保証期間を過ぎた製品に関しては弊社規定による修理費をご請求させていただきます。

修理品の返却

修理品のご返送に際しては、お手数ですが、製品名、製造番号、不良の内容、ご担当者の名前、ご所属、電話番号等を明記ください。返送用の箱は、運送中の事故を避けるため、納入時に使用したダンボール箱、または同等の緩衝材をもつ箱をご使用ください。運送中の事故により修理期間が大幅に長引いたりする場合が考えられますので、適当な梱包箱が見つからない場合は、岩通計測株式会社または岩通ティー・エム・イー・サービス株式会社に必ずご連絡ください。

技術サポート

設置や校正、装置の取り扱いに関するご質問は、次ページまでご連絡下さい。

岩通計測株式会社

岩通ティー・エム・イー・サービス株式会社

サービスネットワーク

本　　社	〒168-8501	東京都杉並区久我山1 - 7 - 4 1	☎(03)5370-5172	FAX(03)5370-5258
西日本営業所	〒541-0054	大阪市中央区南本町3-6-14(イトウビル)	☎(06)6243-4533	FAX(06)6243-4675
会津サービス課	〒965-0855	福島県会津若松市住吉町3　1	☎(0242)26-4339	FAX(0242)26-4348

技術的なお問合わせ窓口（取扱い・測定方法など）

計測フリーダイヤル 0120-086-102　　**会津サービス課フリーダイヤル 0120-267-905**

リモートでオシロスコープを操作

オシロスコープには、フロント・パネル制御部よりマニュアルで行う操作と、外部のコントローラ（通常はコンピュータですが単純なターミナルの場合もあります）よりリモートで行う操作の2つのモードがあります。

本書ではリモート・モードでオシロスコープを操作する方法を説明します。フロント・パネル制御部よりマニュアルで行う操作に関しては、取扱説明書（Operator's Manual）を参照してください。

オシロスコープをリモート・コントロールするには、GPIB（General Purpose Interface Bus）かRS-232-Cの通信ポートを使うことができます。これにより、オシロスコープの電源投入とリモート・アドレスの設定外はすべてリモート・モードでコントロールできます。

この章では、RS-232-CとGPIBに共通する基本的なリモート・コントロールの概念を説明します。また、リモート・コントロール・メッセージについても概説します。

第3、4章では、RS-232-Cインターフェイスや GPIBインターフェイス経由でプログラム・メッセージを送信する方法を説明します。第5章では、波形の転送やフォーマットに関して、その操作方法も含めて詳細に説明します。第6章では、エラー・レポートにステータス・バイトを使用する方法を説明します。

第6章の後のシステム・コマンドのセクションでは、すべてのシステム・コマンドに関して、そのディレクトリも含め詳細に説明します。

GPIB実装標準

このオシロスコープに用いられるリモート・コマンドはGPIB IEEE 488.2[1]標準に準拠しています。この標準は、主に電気及び機械部分を対象としていたIEEE488.1標準の拡張版と見ることができます。IEEE 488.2勧告は、その適用可能な部分はすべてRS-232-C通信にも適用されています。

[1] ANSI/IEEE.488.2-1987 IEE standard codes, (The Insitute of Electronics and Engineers)－電気電子技術者協会の規格 488.2

プログラム・メッセージ

オシロスコープをリモート・コントロールするには、コントローラより、フォーマット構造に従ったプログラム・メッセージを送信する事により行なわれます。これによりオシロスコープでは、正しい形式のプログラム・メッセージはすべて実行され、エラーが検出されると無視されます。

警告メッセージやエラー・メッセージは、コントローラが関連するステータス・レジスタを明示的に調べない限り、あるいはエラー発生時にコントローラから割り込みできるようにステータ

ス・イネーブル・レジスタが設定されていない限り、通常はオシロスコープからはレポートされません。ステータス・レジスタの説明は第6章にあります。

コントロール・プログラム開発中、エラー・メッセージを含めリモート・コントロール・トランザクションはすべて、モニタすることができます。詳細は"COMM_HELP"コマンドを参照下さい。

コマンドとクエリ

プログラム・メッセージは、1つまたは複数のコマンドあるいはクエリで構成されます。コマンドは、たとえばタイムベースや垂直感度などのオシロスコープの状態を変更します。クエリは、オシロスコープにその状態を問い合わせます。コマンドとクエリで同じ文字が頻繁に使われますが、クエリには末尾に<?>がつきます。

たとえば、タイムベースを2 msec/divに変更するには、コントローラよりオシロスコープに次のコマンドを送信します。

```
TIME_DIV 2 MS
```

オシロスコープにタイムベースを問い合わせるには、次のクエリを送信します。

```
TIME_DIV?
```

オシロスコープにクエリが発せられると、オシロスコープより応答メッセージが返されます。このメッセージは、コントローラのGPIBインターフェイスまたはRS-232-Cインターフェイスへの"read"インストラクションでコントロール・プログラムによりリードされます。クエリに対する応答メッセージは、たとえば次のようになります。

```
TIME_DIV 10 NS
```

疑問符の前のクエリ部は、応答メッセージの1部として繰り返されます。この繰り返し部分は、"COMM_HEADER"コマンドで適宜抑止することができます。

コントローラに応答が返されるまでには、オシロスコープの状態と実行される計算により数秒かかることがあります。コマンドの処理はオシロスコープの他の動作に比べて優先度の高いものではありません。したがって、コントローラのIOタイムアウト条件を3秒以上に設定するのが適当です。さらに、クエリ・メッセージに誤りがあると、応答メッセージが返されないことにも注意してください。

ローカル・ステートとリモート・ステート

REMOTE時オシロスコープではリモート・コマンドのみ実行され、クエリも常に実行されます。尚、フロント・パネルに影響を与えないコマンドのうちローカルで実行されるものもいくつかあります（これらのコマンドはシステム・コマンドのセクションのはじめにリストされています）。

REMOTE時オシロスコープでは、LOCALモードに戻すメニュー・ボタンやリモート・コンピュータとの通信に使用するメニュー・ボタンを除き、フロント・パネル・コントロールは無視されます。オシロスコープをLOCAL, REMOTE, LOCAL LOCKOUTに設定する方法に関しては、第3章と4章のGPIBとRS-232-Cをそれぞれ参照してください。

プログラム・メッセージ・フォーム

オシロスコープは、次のようにセミコロン<;>で区切られ終端文字で終わる1つ以上のコマンドやクエリで構成されたプログラム・メッセージでリモート制御されます。

<command/query>;.....;<command/query><終端文字>

プログラム・メッセージには大文字、小文字、あるいはその両方を使用できます。

プログラム・メッセージはオシロスコープの入力バッファ（256 byte）が一杯になった時点を除き、終端文字が受信されるまでは解釈されません。又、コマンドやクエリの実行はそれらの転送順に行われます。

GPIBモードでは、次の終端文字が有効です。

<NL>	改行文字（ASCIIの改行文字であり、10進数表現では10）
<NL><EOI>	同時<EOI>記号付き改行文字
<EOI>	プログラム・メッセージの最終文字につけられる<EOI>記号

注 ： <EOI>記号は、GPIBドライバに特殊なコールを設定する時に用いられるGPIBインターフェイス専用ラインです。GPIBインターフェイス製造者マニュアルとそのサポート・プログラムを参照してください。

<NL><EOI>終端文字は、オシロスコープよりコントローラに送信される応答メッセージに常に用いられます。

RS-232-Cでは、終端文字は"COMM_RS232"コマンドでユーザ定義できます。デフォルト値は<CR>、つまりASCIIの復帰文字、10進数表現で13です。

例

GRID DUAL

このプログラム・メッセージは、オシロスコープにデュアル・グリッド表示を指示する単一コマンドです。終端文字は通常、GPIB（またはRS-232-C）に書き込むインターフェイス・ドライバ・ルーチンによって自動的に付加されるため、表示されていません。

DZOM ON; DISPLAY OFF; DATE?

このプログラム・メッセージは、後ろにクエリが続く2コマンドで構成されます。これは、オシロスコープに対してマルチズーム・モードをオンにして、ディスプレイをオフにするよう指示し、現在の日付を問い合わせています。この場合も終端文字は表示されていません。

コマンド/クエリ・フォーム

コマンドやクエリの一般形式は、コマンド・ヘッダ<header>と、それに続くカンマで区切られたオプションの1つ以上のパラメータ<data>で構成されます。

<header>[?]<data>,...,<data>

ここで表示[?]は、（コマンドをクエリにする）疑問符がオプションであることを示します。コマンドがクエリにもできるかどうかはすべてシステム・コマンドのセクションで詳細に記述されています。ヘッダと後続の最初のパラメータの間にはスペースを1個入れます。パラメータ間にはカンマを入れます。

例

DATE 15,JAN,1993,13,21,16

このコマンドは、オシロスコープに日時を1993年1月15日13時21分16秒に設定するよう指示します。このコマンドのヘッダ"DATE"で動作が規定され、6つのデータ値で詳細が指定されます。

ヘッダ

ヘッダは、オシロスコープで実行される動作のニーモニックです。コマンド・ニーモニックはすべてシステム・コマンドのセクションにアルファベット順にリストされています。
ほとんどのコマンド/クエリ・ヘッダには、最適な判読性を備えたロングフォーマットと、転送と解読を高速にするショートフォーマットがあります。この2形式は完全に同一であり、どちらを使ってもかまいません。たとえば、トリガ・モードを自動に切り替える次の2コマンドは完全に同一です。

TRIG_MODE AUTOとTRMD AUTO

尚、コマンド／クエリには、IEEE 488.2標準に規定されているものもあります。これらが標準化されていることにより、異なる装置間でも類似する動作を同じプログラム・インターフェイスで表現できます。これらのコマンド／クエリにはすべて、その先頭にアスタリスク<*>がつきます。たとえば、"*RST"コマンドは装置をリセットするIEEE 488.2規定の文字列であり、"*TST?"コマンドは装置に自己試験を実行させその結果をレポートさせるIEEE 488.2規定の文字列です。

ヘッダ・パス

コマンドやクエリには、ディスプレイ上の単一入力チャネルやトレースなどオシロスコープのサブ・セクションに適用されるものもあります。この場合、そのコマンドを適用するチャネルやトレースを示すパス名をヘッダの前につける必要があります。ヘッダ・パスは通常、2文字のパス名とその後に続くコロン<:>で構成され、コマンド・ヘッダの直前につけられます。

通常は波形トレースの1つをヘッダ・パスで指定できます（個々のコマンド・ヘッダに適用する値詳細は、システム・コマンドのセクションにリストされているコマンドを参照してください）。

C1、C2	チャネル1、2
C3、C4	チャネル3、4（4チャネル構成時）
M1、M2、M3、M4	メモリ1、2、3、4
TA、TB、TC、TD	トレースA、B、C、D
EX、EX10	外部トリガ
LINE	LINEトリガ・ソース

例

```
C1:OFST  -300 MV
```

チャネルのオフセットを-300 mVに設定します。

ヘッダ・パスの指定は1回だけですみます。後続のコマンドにヘッダ指定がない場合は、最後に指定されたパスを参照するものと見なされます。たとえば、次の2つのコマンドは同一です。

C2:VDIV?;C2:OFST?	チャネル2の垂直感度とオフセットを問い合わせ
C2:VDIV?;OFST?	パス指定を繰り返していない点を除き、上記と同じ

データ

コマンド/クエリにデータ値を追加するときには、その値はASCII文字で指定します。例外は"WAVEFORM"で、波形転送するときにその波形をバイナリ・データ値のシーケンスで指定できます。波形フォーマットの詳細は第5章を参照してください。

ASCIIデータ形式には、文字、数値、文字列、ブロック・データがあります。

文字データ

動作指定の表示には、簡略語や省略語があります。

```
DUAL_ZOOM  ON
```

このデータ値"ON"は、デュアル・ズーム・モードをオンにすることを意味します。

尚、コマンドによっては、いくつものパラメータを指定できる場合、あるいはすべてのパラメータを同時に適用しない場合がありますが、これらのコマンドではペアのデータ値が必要になります。最初のデータ値で変更を加えるパラメータ名を指定し、次のデータ値でその値を指定します。指定は変更を加えるペアのパラメータのみ記述します。

```
HARDCOPY_SETUP  DEV,EPSON,PORT,GPIB
```

上記では2ペアのパラメータが指定されています。最初のパラメータでデバイスをEPSONプリンタに指定し、次のパラメータでGPIBポートを指定しています。"HARDCOPY_SETUP"コマンドではさらに多くのパラメータを指定できますが、それらはプリンタ関連のものでなかったり、あるいは未変更のままでかまわないものです。

数値データ

数値データ・タイプは数量情報の入力に用います。数字には、整数、小数、指数が使用できます。

```
TA:VPOS  -5          トレースAの表示トレースを5デビジョン分下に動かします。
C2:OFST  3.56        チャネル2のDCオフセットを3.56Vに設定します。
TDIV  5.0E-6         タイムベースを5 μsec/divに調節します。
```

注 : 数値には10の整数乗倍を表す接頭語や単位をつけて、数値表現を修飾することができます。この場合は下記文字が使用できます。

EX	1E18	Exa-	PE	1E15	Peta-
T	1E12	Tera-	G	1E9	Giga-
MA	1E6	Mega-	K	1E3	kilo-
M	1E-3	milli-	U	1E-6	micro-
N	1E-9	nano-	PI	1E-12	pico-
F	1E-15	femto-	A	1E-18	atto-

たとえば、いろいろな方法でオシロスコープのタイムベースを5 μsec/divに設定できます。

```
TDIV  5E-6           接尾記号なしの指数表記
TDIV  5 US           秒の接尾単位記号"S"と、10の6乗倍（1E-6）を表す接頭語"U"を
                     表す接続語"U"を付けた場合。
TDIV  5000 NS
TDIV  5000E-3 US
```

文字列データ

このデータ・タイプにより、1つのパラメータで長い文字列を転送できます。文字列データ形式は、シングル・クォーテーション記号またはダブル・クォーテーション記号間に任意のシーケンスのASCII文字を入れて作成します。

```
MESSAGE  'Connect probe to point J3'
```

これにより、オシロスコープ画面のグリッド上のメッセージ・フィールドにこのメッセージが表示されます。

ブロック・データ

これらは、16進数ASCIIにコード化されたバイナリ・データ値です。つまり、4ビットのニブル値が16進数の0、...9、A、...Fに変換されて、ASCII文字として転送されます。これらは、波形の転送（"WAVEFORM"コマンド）とオシロスコープの構成情報の転送（"PANEL_SETUP"コマンド）にのみ用います。

応答メッセージ形式

オシロスコープは、クエリに対する回答としてコントローラに応答メッセージを送信します。このメッセージ・フォーマットはプログラム・メッセージ・フォーマットと同じです。つまり、コマンド・フォーマットの個々の応答は、セミコロン<:>で区切られ、終端文字で終了します。これらはオシロスコープより受信したフォーマットと同じフォーマットで、オシロスコープに返信でき、そのままオシロスコープにより有効コマンドとして受け付けられます。 GPIB応答メッセージでは、<NL><EOI>終端文字が常に用いられます。

たとえば、コントローラより下記のプログラム・メッセージ

TIME_DIV?;TRIG_MODE NORM;C1:COUPLING?（終端文字は表示されない）

を送信すると、オシロスコープより下記の応答が返されます。

TIME_DIV 50 NS;C1:COUPLING D50（終端文字は表示されない）

この場合の応答メッセージはクエリのみを参照し、"TRIG_MODE"は無視されています。又、この応答を、そのままオシロスコープに返信すると、タイムベースを50 nsec/divに、チャネル1の入力カップリングを50Ωに設定するプログラム・メッセージになります。

オシロスコープからの応答を必要とするときは、コントロール・プログラムによりGPIBインターフェイスまたはRS-232-Cインターフェイスに対して、オシロスコープよりデータを読み込むよう指示する必要があります。尚、以前のプログラム・メッセージに対する応答を読み込む前に別のプログラム・メッセージを送信すると、オシロスコープの出力バッファに入れられている応答メッセージは廃棄されます。

オシロスコープは、プログラム・メッセージ受信よりも応答メッセージに厳しいルールを適用します。コントローラから送信するプログラム・メッセージには大文字や小文字を使用できますが、応答メッセージは常に大文字で返されます。プログラム・メッセージには余分なスペースやタブ（ホワイト・スペース類）を混在できますが、応答メッセージにはこれらは含まれません。プログラム・メッセージにはショート・フォーマットとロング・フォーマットのコマンド/クエリ・ヘッダを混在できますが、応答メッセージでは常にデフォルトのショートフォーマットのヘッダが使われます。しかし"COMMAND_HEADER"コマンドを用いると、ロングフォーマットのヘッダを使うか、あるいはヘッダを全く付けないかを、オシロスコープに設定することができます。応答ヘッダを省略すると応答転送時間を最短化できますが、この応答はオシロスコープに再送できません。さらにこの応答には接尾単位記号も取り除かれます。

たとえば、チャネル1のトリガ・スロープが負に設定されているときに"C1:TRSL?"クエリを発すると下記いずれかの応答が返されます。

C1:TRIG_SLOPE NEG	ヘッダ・フォーマット: ロング
C1:TRSL NEG	ヘッダ・フォーマット: ショート
NEG	ヘッダ・フォーマット: オフ

"WAVEFORM"クエリでオシロスコープより波形データを取り込むと、その応答メッセージは特殊なフォーマットで返されます。これらのフォーマットは"COMM_FORMAT"コマンドと"COMM_ORDER"コマンドで厳密に指定できます。

GPIBを用いる操作

この章では、汎用インターフェイス・バス(GPIB: General Purpose Interface Bus)でオシロスコープをリモート制御する方法を説明します。インターフェイス機能、アドレス指定方式、標準バス・コマンド、ポーリング・スキーマについても説明します。

GPIB構造

GPIBは通常のコンピュータ・バスに似ています。両者の違いは、コンピュータではバックプレーン・バス経由で相互接続されたバスボードがあるのに対して、GPIBではケーブル・バスを介して各デバイスが独立に相互接続される点です。尚、両方のバスともプログラム・メッセージとインターフェイス・メッセージが搬送されます。

プログラム・メッセージはしばしばデバイス依存メッセージとも呼ばれますが、これには、プログラミング命令、測定結果、測定器ステータス、波形データが含まれます。これらの一般フォーマットは第2章に説明があります。

インターフェイス・メッセージはバスの管理を行います。これらは、バスのイニシャライズ、デバイスのアドレス割当と解除、リモート・モードとローカル・モードの設定などの機能を遂行します。

GPIB上のデバイスは、リスナ（listener）か、トーカ（talker）またはコントローラかその両方になれます。トーカは複数のリスナにプログラム・メッセージを送信します。コントローラは、インターフェイス・メッセージを各デバイスに送信してバス上の情報の流れを管理します。オシロスコープはトーカにもリスナにもなれますが、コントローラにはなれません。しかし、ホスト・コンピュータはこれら3つのモードを実行できなくてはなりません。コントローラがGPIBを特定機能に構成する方法詳細は、GPIBインターフェイス製造者提供のマニュアルを参照してください。

インターフェイス機能

オシロスコープのインターフェイス機能には、IEEE 488.1で定義された下記の機能があります。

AH1	全てのアクセプタ・ハンドシェーク
SH1	全てのソース・ハンドシェーク
L4	一部のリスナ・ファンクション
T5	全てのトーカ・ファンクション
SR1	全てのサービス要求ファンクション
RL1	全てのリモート／ローカル・ファンクション
DC1	全てのデバイス・クリア・ファンクション
DT1	全てのデバイス・トリガ

PP1　　パラレル・ポーリング・リモート
C0　　コントローラ・ファンクションなし
E23　　トライステート・ドライバ

アドレス指定

GPIB上のデバイスはすべてアドレスを持っています。オシロスコープは、("UTILITIES"メニューで）リモート・コントロール・ポートを"GPIB"に設定すると、GPIBで制御できます。リモート・コントロール・ポートを"RS-232"に設定しているとき、GPIB上ではオシロスコープは "トークオンリー" 動作（プリンタ駆動など）になります。オシロスコープを"UTILITIES"メニューで"RS-232"に設定しているとき、RS-232-Cポートからは、そのオシロスコープを制御できます。

オシロスコープのアドレスをトークに指定すると、ユニバーサル・アントーク（UNT）コマンド、オシロスコープ自体のリスンアドレス（MLA）、あるいは他の測定器のトーク・アドレスが受信されるまではトーク構成のままになります。

同様に、オシロスコープのアドレスを、リスンに指定するとユニバーサル・アンリスン（UNL）コマンド、あるいはオシロスコープ自体のトーカ・アドレス（MTA）が受信されるまではリスン構成のままになります。

GPIB信号

このバス・システムは、16本の信号ラインと8本のグランド（GND）ライン（またはシールドライン）で構成されます。信号ラインはさらに次の3グループに区分けされます。

― データ・ライン8本
― ハンドシェイク・ライン3本
― インターフェイス・コントロール・ライン5本

データ・ライン

通常、8本のデータ・ラインはDIO1～DIO8と呼ばれ、プログラム・メッセージとインターフェイス・メッセージの搬送に用いられます。尚、ほとんどのメッセージには7ビットASCIIコードが用いられますので、この場合はDIO8は使用されません。

ハンドシェイク・ライン

この3つのハンドシェーク・ラインは、デバイス間のメッセージ・バイト交換の制御に用いられます。このプロセスは3ライン・インターロック・ハンドシェイクと呼ばれ、これによりデータライン上のメッセージ・バイトが伝送エラーなしに送受信されることが保証されています。

インターフェイス・コントロール・ライン

下記の5本のラインで、インターフェイス上の情報の流れが管理されます。

アテンション（ATN : ATteNtion）

コントローラは、トーク・アドレスやリスン・アドレスなどのメッセージやデバイス・クリア（DCL）メッセージの送信でデーラインを使用するとき、ATNをTRUEにします。尚、ATNがFALSEのときはバスはデータ・モードになっているので、トーカからリスナへプログラム・メッセージを転送できます。

インターフェイス・クリア（IFC : InterFace Clear）

コントローラは、バスをイニシャライズするときIFCを一定時間TRUEにします。

リモート・イネーブル（REN : Remote ENable）

コントローラは、デバイスをリモート・モードやローカル・モードにするときこのラインを使用します。

サービス・リクエスト（SRQ : Service ReQuest）

どのデバイスもSRQをTRUEにして、コントローラに同期する事なくサービスを要求できます。このラインは、コンピュータ・バスの単１インタラプト・ラインと同一です。

終端または識別（EOI : End Or Identify）

このラインには2つの用途があります。トーカはこのラインを使用してメッセージ・ストリングの終りを示します。コントローラはこのラインを使用してデバイスに、パラレル・ポーリングでそれぞれの応答を識別するよう指示します（このセクションの後方で説明します）。

I/Oバッファ

オシロスコープには256バイトの入力バッファと出力バッファがあります。受信されたプログラム・メッセージは、メッセージ終端文字が受信されるまでは解読されません。しかし、（プログラム・メッセージが長すぎて）入力バッファが満杯になると、オシロスコープはメッセージを解読し始めます。この場合、データ転送は一時的に中断され、タイムアウトの設定を極端に短く設定していると、コントローラによりタイムアウトを起こすことがあります。

IEEE 488.1標準メッセージ

IEEE 488.1標準では、GPIBの機械電気系ばかりでなく下位層の転送プロトコルも規定しています。たとえば、コントローラがデバイスをアドレス指定し、トーカやリスナに切り替え、リモート・ステートをリセットしたりリモート・ステートに設定したりする方法が規定されています。このようなインターフェイス・メッセージは、GPIBのインターフェイス管理ラインで（通常はATNをTRUEにして）送信されます。

これらのメッセージはすべて（GETを除いて）受信直後に実行され、通常のコマンドのように入力順には実行されません。

注1 : IEEE 488.1インターフェイス・メッセージ標準に加え、IEEE 488.2標準にはコマンド・ヘッダなど標準化されたプログラム・メッセージがいくつか規定されています。これらは、先頭にアスタリスク<*>を付けて識別され、システム・コマンドのセクションにリストされています。

システム・コマンドのセクションには、入出力の各バッファをクリアするコマンドやオシロスコープをリモート・ステートに設定するコマンドはリストされていません。これは、このようなコマンドはIEEE 488.1標準メッセージに既に規定されているからです。ホスト・コンピュータのGPIBインターフェイス・マニュアルとそのサポート・プログラムを参照してください。それらには、これらのメッセージを実行する特別なコールが記述されているはずです。

以下に、単にバスの再構成を行うだけでなくオシロスコープの動作に影響を与えるIEEE 488.1標準メッセージを説明します。

デバイス・クリア

Device CLear （DCL）またはSelected Device Clear（SDC）に対しては、オシロスコープは入力バッファと出力バッファをクリアし、たとえ現在解読中のコマンドがあったとしてもコマンド解読作業を中止し、保留中のコマンドはすべてクリアしますが、ステータス・レジスタとステータス・イネーブル・レジスタはクリアされません。DCLは即時に実行されますが、オシロスコープが他の動作を行っている場合などにはこのコマンドが実行されるまでに数秒かかることがあります。

グループ実行トリガ

Group Execute Trigger（GET）メッセージにより、オシロスコープのトリガ・システムが機能します。これは、"*TRG"コマンドと機能的に同じです。

リモート・イネーブル

コントローラによってRemote ENableコントロールラインがTRUEに保持され、オシロスコープがリスナとして指定されると、このインターフェイス・メッセージが実行されます。この時、メニュー・ボタンを除くフロント・パネル制御部はすべて無視されます。又、画面右端のメニューは使用できないため、表示されません。代わりに、REMOTE ENABLEテキストがメニュー・フィールドの最上部に表示され、オシロスコープがリモート・モードに設定されていることを示します。

コントローラによりRENラインがFALSEに戻されると、そのバス上のオシロスコープはすべてLOCALに戻されます。個々のオシロスコープをLOCALに戻すには、Go To Localメッセージ（ローカル移行を参照してください）を用います。

ルール上、クエリは常に実行されますが、リモート・コマンドはオシロスコープがリモート・ステートのときにのみ実行されます。オシロスコープがローカル・ロックアウト（LLO: Local

LOckout）モードに設定されていない限り、LOCALメニュー・ボタンを押すことでローカルのフロント・パネル制御に戻せます。

ローカル・ロックアウト

Local LOckout（LLO）コマンドは、LOCALメニューを見えなくします。LLOコマンドはローカル・モードでもリモート・モードでも送信できますが、オシロスコープがリモート・モードに設定されているときにのみ有効になります。

ローカル移行

Go To Local（GTL）メッセージは、オシロスコープをローカル・モードに戻します。フロント・パネル制御がすべてアクティブになり、通常のメニューが再表示されます。その後、オシロスコープはリスナに設定されると、直ちにリモート・ステートに戻されます。

ローカル・ロックアウトが設定されているときは、GTLメッセージによってもそのローカル・ロックアウトはクリアされないことに注意してください。したがって、オシロスコープがリモート・ステートに戻されると即座にローカル・モードが再び有効になります。

コマンド文字列の後ろに直接、GTLメッセージを並べないでください。GTLは直ちに実行されるので、入力バッファ中のコマンド解読前に、オシロスコープが既にローカル・ステートに戻ることがあります。この場合、オシロスコープをREMOTEに戻すコマンドを送信したとしてもオシロスコープ側で実行されません。コマンドがすべて解読されるようにするためには、コマンド文字列にクエリを追加し（たとえば、"*STB?"）、その応答を待ってからGTLメッセージを送信します。

インターフェイス・クリア

InterFace Clear（IFC）メッセージは、GPIBをイニシャライズしますが、オシロスコープの動作にはなにも影響を及ぼしません。

GPIB転送のプログラミング

GPIBプログラミング概念については、このセクションにBASICAで記述された数多くの例で説明されています。ここでは、コントローラはDOSを搭載したIBM-PC互換機であり、ナショナル・インスツルメンツ[1]のGPIBインターフェイス・ボードが装着されているものを想定しています。CやPascalなどの他の言語によるGPIBプログラミングも、このプログラミングに極めて似ています。他のコンピュータや他のGPIBインターフェイスをご使用のときは、インターフェイス・マニュアルを参照して、ここに説明されているものと同じようにしてインストレーション手順やサブルーチン・コールを実行してください。

[1] National Instruments Corporation, 日本ナショナル・インスツルメンツ株式会社

GPIBハードウェアの構成

GPIBインターフェイスがコンピュータに正しくインストールされているかチェックします。正しくインストールされていないときは、そのインターフェイス製造者のインストレーション手順に従って再インストールします。ナショナル・インスツルメンツのインターフェイスボードの場合は、スイッチとジャンパで、そのボードのベースI/Oアドレス、DMAチャネル番号、割り込み線設定を変更できます。当社のプログラム例では、デフォルトの位置設定を想定しています。（Windows95ではプラグ＆プレイで使用できるインターフェイスも有ります）

オシロスコープとコンピュータにGPIBインターフェイス・ケーブルを接続します。GPIBアドレスを必要な値に設定します。プログラム例では、設定4を想定しています。

GPIBドライバ・ソフトウェアの構成

ホスト・コンピュータには、オペレータのプログラムとインターフェイス・ボード間でトランザクション処理するインターフェイス・ドライバが必要です。ナショナル・インスツルメンツのインターフェイスの場合は、下記の手順に従ってインストールします。

- GPIBハンドラGPIB.COMをブート・ディレクトリにコピーします。
- DOSシステムのコンフィギュレーション・ファイルCONFIG.SYSを変更して、GPIBハンドラの存在を宣言します。
- サブディレクトリGPIB-PCを作成します。
- システムのテストと再構成およびユーザ・プログラムの記述に有用なファイルやプログラムをGPIB-PCサブディレクトリにインストールします。

GPIB-PCサブディレクトリにインストールするファイルを以下に示します。

IBIC.EXE

これにより、キーボードより関数を入力してGPIBを対話的に制御できます。GPIBプログラミングやオシロスコープのリモート・コマンドに不慣れなユーザには、このプログラムの使用を推賞します。

DECL.BAS

これは、BASICAアプリケーション・プログラムの先頭に入れるコードを集めた宣言ファイルです。DECL.BASにオペレータの命令を追加し、コンパイル・ファイルを実行するだけで、簡単なアプリケーションを迅速に記述できます。

IBCONF.EXE

これは、GPIBハンドラの現在の設定値の検査や変更を可能にする対話型プログラムです。IBCONF.EXEを実行するにはナショナル・インスツルメンツのマニュアルを参照してください。このセクションのプログラム例では、ナショナル・インスツルメンツのGPIBドライバGPIB.COMがデフォルト・ステートに設定されている、つまり、ユーザがIBCONF.EXEでこのドライバ設定を変

更していないことを想定しています。これは、インターフェイス・ボードがシンボリック名'GPIB0'で参照でき、GPIBバスのアドレス1～16のデバイスがシンボリック名'DEV1'～'DEV16'でコールできることを意味しています。

注2 : お手持ちのナショナル・インスツルメンツ・インターフェイス・カードがPC2Aではなく PC2の場合は、IBCONF.EXEを実行して、デフォルトのPC2Aではなく PC2の存在を宣言してください。

簡易転送

ナショナル・インスツルメンツの提供する3つのサブルーチン（IBFIND、IBRD、IBWRT）を用いるだけで、膨大な数のリモート・コントロール操作を実行できます。下記のプログラムだけで、オシロスコープのタイムベース設定を読み出し、端末に表示できます。

```
1-99        <DECL.BAS>
100         DEV$="DEV4"
110         CALL IBFIND (DEV$, SCOPE%)
120         CMD$="TDIV?"
130         CALL IBWRT (SCOPE%, CMD$)
140         CALL IBRD (SCOPE%, RD$)
150         PRINT RD$
160         END
```

説　明

1～99行は、ナショナル・インスツルメンツの提供するDECL.BASファイルのコピーです。先頭の6行は、GPIBハンドラのイニシャライズに必要です。他の行は、大きなプログラムには有用な宣言部ですが、実行コードには必要ありません。上記のプログラム例では、255文字の配列としてDECL.BASに宣言されているCMD$文字列とRD$文字列のみ使われています。

注3 : GPIBイニシャライズ時、DECL.BASはBIB.Mファイルをアクセスする必要があります。BIB.Mは、ナショナル・インスツルメンツの提供するファイルの1つですが、カレント・ディレクトリに入れておく必要があります。

注4 : DECL.BASの先頭の2行には各行に"XXXXX"文字列がありますが、これは、（BASICAで現在使用可能な空間よりBIB.Mサイズを差し引いて計算される）BASICAの最大ワークスペースを決定するバイト数で置き換える必要があります。たとえば、BIB.Mサイズが1200バイトで、BASICAロード時"60200 bytes free（バイト空き）"と表示された場合は、"XXXXX"の所に5900以下の値を入れます。

行100と行110は、"DEV4"をオープンし、これを記述子"SCOPE%"に関連づけます。これ以降、I/Oコールはすべて"SCOPE%"を参照することになります。GPIBハンドラのデフォルト・コンフィギュレーションでは、"DEV4"が認識され、GPIBアドレス4のデバイスに関連づけられます。

行120と130は、コマンド文字列TDIV?を用意し、これをオシロスコープへ転送します。このコマンドは、オシロスコープにタイムベースの現在設定値を返すよう命令します。

行140と150は、オシロスコープの応答を読み込み、それを文字列RD$に代入します。

行170は、端末に応答を表示します。

このプログラム例実行時、IBWRTが実行されるとオシロスコープは自動的にリモート・ステートにおかれ、その状態を保ち続けます。GPIBハンドラがローカル・ロックアウト（LLO）禁止に設定されてれば、LOCALメニュー・ボタンを押すと、オシロスコープはローカル・モードに戻ります。

ここで、プログラム例を少し変更して、GPIB操作時に発生したエラーをチェックするプログラムを紹介します。

```
1-99        <DECL.BAS>
100         DEV$="DEV4"
110         CALL IBFIND (DEV$, SCOPE%)
120         CMD$="TDIV?"
130         CALL IBWRT (SCOPE%, CMD$)
140         IF ISTA%<0 THEN GOTO 200
150         CALL IBRD (SCOPE%, RD$)
160         IF ISTA%<0 THEN GOTO 250
170         PRINT RD$
180         IBLOC (SCOPE%)
190         END
200         PRINT "WRITE ERROR=";IBERR%
210         END
250         PRINT "READ ERROR=";IBERR%
260         END
```

GPIBステータス・ワードISTA%、GPIBエラー変数IBERR%、カウント変数IBCNT%はGPIBハンドラで定義されており、GPIBファンクション・コールのたびに更新されます。詳細はナショナル・インスツルメンツのマニュアルを参照してください。上のプログラム例では、オシロスコープのGPIBアドレスが4以外の値に設定されているとレポートされます。行180は、GPIBルーチンIBLOCをコールしてオシロスコープをローカルに戻します。

いくつかの追加ドライバ・コール

IBLOC

IEEE 488.1標準のGTLメッセージを実行します（つまり、オシロスコープをローカル・ステートに戻します）。上記のプログラム例でも使われています。

IBCLR

IEEE 488.1標準のSDCメッセージを実行します

IBRDFとIBWRTF

IBRDFでGPIBよりファイルへのデータ読み込みが、IBWRTFでファイルからGPIBへのデータ書き込みが行なわれます。保存装置との直接データ転送にはデータ・ブロック・サイズの制限はありませんが、コンピュータ・メモリとのデータ転送に比べると遅くなります。

IBRDIとIBWRTI

IBRDIでGPIBより整数配列へのデータ読み込みが、IBWRTIで整数配列からGPIBへのデータ書き込みが可能になります。（BASICでは）整数配列記憶が最大64Kバイトなので、大容量のデータ・ブロックをコンピュータに転送するときにはBASIC文字列長で256バイトに制限されているIBRDFやIBWRTFではなくIBRDIとIBWRTIを用いる必要があります。Cなどの最近のプログラム言語ではIBRDとIBWRTファンクション・コールのデータ・ブロック・サイズの制限は遥かに軽減されているので、IBRDIとIBWRTIはBASIC用のみ用意されていることに注意してください。

IBTMO

プログラム実行時にタイムアウト時間を変更します。GPIBドライバのデフォルト値は10秒です。たとえば、オシロスコープがIBRDコールに応答しないとき、IBRDは指定時間後にエラーを返します。

IBTRG

IEEE 488.1標準のGETメッセージを実行します。これによりオシロスコープはトリガ待ち状態になります。

ナショナル・インスツルメンツは追加ファンクション・コールを数多く提供しています。特に、GPIBを極めて詳細に制御できる、いわゆるボード・レベル・コールを使用できます。

サービスリクエストのプログラミング

オシロスコープをリモート・アプリケーションで使用していると、しばしば非同期にイベントが発生します。つまり、その時点はホスト・コンピュータからは予測できません。最も一般的な例は、オシロスコープがトリガ待ち状態になっている場合です。コントローラは、捕捉が終了す

るまで待機していないと、捕捉波形を読み込むことができません。あるイベントが発生したがどうかをチェックする簡単な方法は、それに関連するステータス・ビットを、求める遷移状態が検出されるまで継続的または周期的に読み出すことです。継続ステータス・ビット・ポーリングの詳細は、本章のサブセクション "オシロスコープ・ポール"に説明があります。ステータス・バイト詳細は第6章を参照してください。

オシロスコープで発生するイベントを検出する効果的な方法はService Request（SRQ）を用いることです。このGPIB割り込みラインは、コントローラのプログラム実行割り込みに使用できます。したがって、コントローラは他のプログラムを実行しながら、割り込みを待っていることができます。残念ながら、すべてのインターフェイス製造者が割り込みサービス・ルーチンをサポートしているわけではありません。特に、ナショナル・インスツルメンツでは、ISTA%ステータス・ワード内でのSRQビットのみサポートしていますので、IBWAITファンクション・コールを用いて、これらを頻繁にチェックする必要があります。尚、実際に割り込みサービス・ルーチンがサポートされていないところでは、SRQを使用してもあまり有効ではありません。

デフォルト・ステートでは、電源投入後のサービス・リクエストはディスエーブルです。SRQをイネーブルにするには、"*SRE"コマンドでサービス・リクエスト・イネーブル（Service Request Enable）レジスタを設定し、イベントでSRQが生成されるように指定します。オシロスコープは、選択されたイベントが発生すると即座にSRQラインに信号を出し、コントローラに割り込みをかけます。GPIBにデバイスを複数接続しているときは、コントローラは、複数のデバイスを連続ポーリングして、どのデバイスによって割り込みがかけられたかを特定する必要があります。

注5 : SRQビットは、コントローラがステータス・バイト・レジスタ（STatus Byte Register）を読み込むまでラッチされます。"*STB?"コマンドでSTB読み込みを行うと、MAVビット（ビット4）を除きレジスタの内容がクリアされます。その後新しいイベントが発生すると設定されます。SREレジスタをクリアする（"*SRE 0"）と、サービス・リクエストもディスエーブルできます。

例1

"new signal acquired"または"return-to-local"のイベントに対する応答でSRQ信号を生成します。

これらのイベントはINRレジスタに記録され、INRレジスタはSREレジスタの位置0のINBサマリ・ビットに反映されます。ビット位置0には値1があるため、INBサマリ・ビットが設定されているときは、"*SRE 1"コマンドでSRQ生成をイネーブルにします。

さらに、INBビットにサマリ可能なINRレジスタのイベントを指定します。"new signal acquired"イベントはINEビット0（値1）に対応し、"return-to-local"イベントはINEビット2（値4）に代入されます。合計は1+4=5となります。したがって、"INE 5"コマンドを使用します。

```
CMD$="INE 5;*SRE 1"
CALL IBWRT (SCOPE%, CMD$)
```

例2

ソフト・キー4を押してSRQ信号を生成します。

"soft key 4 pressed"イベントはURRレジスタに記録されます。URRレジスタはSTBに直接には反映されずESRレジスタ（URR、ビット位置6）にのみ反映されるため、最初に"*ESE 64"コマンドでESEイネーブル・レジスタを設定し、URQ設定がSTBに記録されるようにします。これにより、SREイネーブル・レジスタのESBサマリ・ビット（ビット位置5）を設定する（"*SRE 32"）とSRQ要求が生成されます。

```
CMD$="*ESE 64;*SRE 32"
CALL IBWRT (SCOPE%, CMD$)
```

インストルメント・ポール（Instrument poll）

オシロスコープ（インストルメント）内で発生するステート遷移は、選択した内部ステータス・レジスタをポーリングしてリモート監視できます。このサブセクションでは、下記に示す、ある与えられたイベントの出現検出に使用できるポーリング方式を説明します。

— 連続ポール（Continuous poll）
— シリアル・ポール（Serial poll）
— パラレル・ポール（Parallel poll）
— *ISTポール（*IST poll）

これらの方式の差異を強調するために、それぞれの説明に、新しい捕捉があったことを判定する同じ例を用います。最も単純なポールは連続ポールです。他の方式は、（SRQ線をサービスする）割り込みサービス・ルーチンがサポートされているか、GPIB上で複数のデバイスを同時に監視する必要のある場合にのみ意味を持ちます。

連続ポール

連続ポーリングでは、遷移が検出されるまでステータス・レジスタが連続的に監視されます。これは、ステート変更を検出する最も直接的な方式ですが、特に複数のデバイス構成のときなど状況によっては実装できな場合があります。

以下の例では、内部ステート変更レジスタ（INR: INternal state change Register）に対して、対応するビット（この例ではビット0、つまり値1）が非0になるまで、つまり新しい波形が捕捉されたことを示すまで連続ポーリングされて、"new signal acquired"イベントが検出されます。INRは読み出されると同時にクリアされ、非0値の検出後にさらにクリアする必要はありません。"CHDR OFF"コマンドは、オシロスコープに対して、クエリへの応答時コマンド・ヘッダはすべて削除するよう指示する命令です。これにより応答の解読が簡単になります。したがって、オシロスコープは"INR 1"ではなく"1"を送信することになります。

```
CMD$="CHDR OFF"
CALL IBWRT (SCOPE%, CMD$)
MASK%=1 'New Signal Bit has value 1'
LOOP%=1
WHILE LOOP%
    CMD$="INR?"
    CALL IBWRT (SCOPE%, CMD$)
    CALL IBRD (SCOPE%, RD$)
    NEWSIG%=VAL (RD$)  AND MASK%
    IF NEWSIG%=MASK% THEN LOOP%=0
WEND
```

シリアル・ポール

SRQラインに信号が出されていると、シリアル・ポールが実行されます。コントローラは、装置ごとのSTBレジスタのSRQビットを調べて、どの装置が割り込みを生成したかを調べます。サービス・リクエストは割り込み機構をベースにしているため、複数デバイス構成時のサービス・スピードについては、シリアル・ポーリングにより妥当な線が得られます。

下記の例では、"INE 1"コマンドで"new signal acquired"イベントをイネーブルにし、INRのステータス・ビットSTBのINBビットに対してレポートされるようにしています。また、"SRE 1"コマンドでステータス・バイトのINBをイネーブルにし、SRQをいつでも設定して生成できるようにしています。IBWAITファンクション・コールでコンピュータに対して、次の3つの状態のどれかが発生するまで待機するように命令しています。GPIBエラーに対応するマスク（MASK%）の&H8000、タイムアウト・エラーに対応する&H4000、SRQビットで生成されたサービス・リクエスト（RQS: ReQuest for Service）の検出に対応する&H0800。

IBWAITでRQSが検出されると、シリアルポールが自動的に実行され、どの装置が割り込みを生成したか調べられます。この動作は、このときタイムアウトであったり、"SCOPE%"装置がSRQを生成していると、単に終了します。さらにIBRSPファンクション・コールで、ステータス・バイト値をフェッチして解析を深めます。この例では、正しく機能させるためには、GPIBハンドラの'Disable Auto Serial Polling'値を、'off 'に設定する必要があります（IBCONF.EXEでチェックします）。

```
CMD$="*CLS; INE 1; *SRE 1"
CALL IBWRT (SCOPE%, CMD$)
MASK%=&HC800
CALL IBWAIT (SCOPE%, MASK%)
IF (IBSTA% AND &HC000) <>0 THEN PRINT "GPIB or Timeout Error":STOP
CALL IBRSP (SCOPE%, SPR%)
PRINT "Status Byte=.", SPR%
```

注6 : シリアル・ポールが完了すると、STBステータス・レジスタのRQSビットはクリアされます。他のSTBレジスタ・ビットは、"CLS"コマンド、または装置リセットによりクリアされるまで設定されたままなので注意してください。これらのビットをクリアしないと、他の割り込みは生成できません。
シリアルポーリングは、複数の装置を監視する必要のある場合にのみ意味を持ちます。ボード・レベルのファンクション・コールで、同一インターフェイス・カードに接続した複数の装置を同時に監視できます。詳細は、ナショナル・インスツルメンツのマニュアルを参照してください。

パラレル・ポール

パラレル・ポーリングもシリアル・ポーリング同様、複数の装置を監視する必要のある場合にのみ意味を持ちます。

パラレル・ポーリングでは、コントローラは、すべての装置の個別ステータス（IST: Individual STatus）ビットを同時に読み出して、どの装置にサービスが必要かを判定します。パラレルポーリングは装置を同時に8台までポーリングできるため、この方式をサポートする各装置のステート変更を特定する最速な方法です。

パラレルポールを起動すると、装置はそれぞれ、DIOデータラインの1ライン上にステータス・ビットを返します。デバイスによっては、別々のDIOラインを使ってパラレルに応答するものや、1本のデータラインでシリアルに応答するものがあります。データライン割り当ては、パラレルポール構成（PPC: Parallel Poll Configure）シーケンス経由でコントローラが行います。

下の例では、"INE 1"コマンドでINRの"new signal acquired"イベントをイネーブルにして、STBステータス・バイトのINBビットにレポートされるようにしています。パラレルポール・イネーブル（PRE: Parallel poll Enable）レジスタで、どのイベントがISTステータス・ビットにサマライズされるか決定されます。"*PRE 1"コマンドでINBビットをイネーブルにして、ISTビットをいつでも設定できるようにしています。パラレルポーリングは確立されると、DIO2データ・ラインに変化が見られるまでパラレルポール・ステータスが調べられます。

ステージ1 : INEレジスタとPREレジスタをイネーブルにし、コントローラをパラレルポールに構成し、オシロスコープにDIO2上に応答するよう命令します。

```
CMD1$="?_@$"
CALL IBCMD (BRD0%, CMD1$)
CMD$="INE 1;*PRE 1"
CALL IBWRT (BRD0%, CMD$)
CMD4$=CHR$ (&H5) +CHR$ (&H69) +"?"
CALL IBCMD (BRD0%, CMD4$)
```

ステージ2 ： DIO2が設定されるまでオシロスコープをパラレルポールします。

```
LOOP%=1
WHILE LOOP%
CALL IBRPP（BRD0%, PPR%）
IF （PPR% AND &H2）=2 THEN LOOP%=0
WEND
```

ステージ3 ： パラレルポーリングをディスエーブルして（16進：15）、パラレルポール・レジスタをクリアします。

```
CMD5$=CHR$（&H15）
CALL IBCMD（BRD0%, CMD5$）
CALL IBCMD（BRD0%, CMD1$）
CMD$="*PRE 0"
CALL IBWRT（BRD0%, CMD$）
```

注7 ： 上記の例では、ボード・レベルのGPIBファンクション・コールが用いられています。コントローラ（ボード）とオシロスコープ（デバイス）はそれぞれアドレス0と4が割り当てられていると想定しています。コントローラとオシロスコープのリスナ・アドレスとトーカ・アドレスを下記に示します。

論理デバイス	リスナ・アドレス	トーカ・アドレス
コントローラ	32（ASCII<space>	64（ASCII @）
オシロスコープ	32+4=36（ASCII $）	64+4=68（ASCII D）

注8 ： コマンド文字列内の"?"と"_"はそれぞれアンリスン（UNL）とアントーク（UNT）を表します。これらは、各デバイスを"既知（known）"ステートに設定するときに用います。

注9 ： プログラム例ではサイズを小さくするために、デバイスのトークとリスンの初期化命令が文字列内にグループ化されています。これらを以下に示します。
CMD1$="?_@$" アンリスン、アントーク、PCトーカ、DSOリスナ

注10: バイナリ形式でパラレル応答を実行するリモート・メッセージ・コードは、01101PPPです。ここで、PPPにはデータラインを指定します。ここではデータライン2が選択されているので、識別コードは001です。これは結果的にコード01101001（バイナリ）あるいは&H69（16進）になります。詳細は、IEEE 488-1978標準の表38を参照してください。

*ISTポール

パラレル・ポーリングで返される個別ステータス（IST: Individual STatus）ビットのステートは、"*IST?"クエリを送信して読み出すことができます。このポール・モードをイネーブルにするには、PREレジスタに書き込んでオシロスコープをパラレル・ポーリング向けにイニシャライズする必要があります。*ISTポーリングはパラレル・ポーリングをエミュレートするので、コントローラによってパラレル・ポーリングがサポートされていなくてもこの方式を用いることができます。

下記の例では、"INE 1"コマンドでINRの"new signal acquired"イベントをイネーブルにして、STBステータス・ビットのINBビットにレポートされるようにしています。"*PRE 1"コマンドでINBビットをイネーブルにして、ISTビットをいつでも設定できるようにしています。"CHDR OFF"コマンドでオシロスコープの応答のコマンド・ヘッダを取り除き、解読を簡単にしています。これにより、ISTビットのステータスが、オシロスコープに設定されるまで連続的に監視されます。

```
CMD$="CHDR OFF; INE 1;*PRE 1"
CALL IBWRT (SCOPE%, CMD$)
LOOP%=1
WHILE LOOP%
    CMD$="*IST?"
    CALL IBWRT (SCOPE%, CMD$)
    CALL IBRD (SCOPE%, RD$)
    IF VAL (RD$) =1 THEN LOOP%=0
WEND
```

ハードコピー・デバイスの駆動

オシロスコープは、広い範囲のハードコピー・デバイスにインターフェイス接続でき、画面内容をそれらに直接コピーするよう命令できます。装置にサポートされているデバイスが、システム・コマンドの"HARDCOPY_SETUP"コマンドでリスンされます。

ハードコピー・デバイスをGPIBに接続するときは、GPIBコントローラを使用できるかできないかにより、下記のいずれかの構成を選択する必要があります。

GPIBコントローラなしの印刷

オシロスコープとハードコピー・デバイスのみGPIBに接続しているときは、オシロスコープをトーカ専用に、ハードコピー・デバイスをリスナ専用にし、データ転送が適切に行われるようにする必要があります。オシロスコープをトーカ専用にするには、"UTILITIES"メニューでリモート・コントロール・ポートをRS-232に設定します。ハードコピー・デバイスをリスナ専用にするには通常、そのハードコピー・デバイス製造者によって指示されているアドレスを指定します。

1. ("UTILITIES"メニューで) リモート・コントロールをRS-232に設定します。

2. "UTILITIES"メニューの"Hardcopy Setup"サブメニューでハードコピー・ポートに"GPIB"を指定します。
3. ハードコピー・デバイスをリスナ専用モードに設定します。
4. オシロスコープのフロント・パネルにあるスクリーン・ダンプ・ボタンを押します。

GPIBコントローラで印刷

GPIBにコントローラを接続しているときは、そのコントローラでデータ転送を管理します。次に示す複数のスキーマで画面内容を転送できます。

1. コントローラがデータを内部メモリに読み込んでからプリンタに送信します。これ以外の方法でも、簡単な高レベルGPIBファンクション・コールで印刷できます。
2. オシロスコープがデータをコントローラとプリンタの両方に送信します。
3. コントローラがスタンバイ・モードになります。オシロスコープがトーカになり、データを直接プリンタに送信します。

1. コントローラがデータを読み込んでからプリンタに送信

コントローラはプリンタ命令フルセットを記憶し、これらをグラフィック・デバイスに送信します。この方式は画面内容の転送には最も直接的ですが、大容量のバッファ記憶装置が必要です。

```
CMD$="SCDP"
CALL IBWRT (SCOPE%, CMD$)
FILE$="PRINT.DAT"
CALL IBRDF (SCOPE%, FILE$)
CALL IBWRTF (PRINTER%, FILE$)
```

2. オシロスコープがデータをコントローラとプリンタの両方に送信

オシロスコープがプリンタ命令をバス上に送出します。データは直接送出され、コントローラのスクラッチ・メモリに格納されます。後に、スクラッチ・ファイルの内容は削除できます。

ステージ1 : コントローラをトーカに、オシロスコープをリスナにして、スクリーン・ダンプ・コマンドを発行します。

```
CMD1$="?_@$":CALL IBCMD (BRD0%, CMD1$)
CMD$="SCDP":CALL IBWRT (BRD0%, CMD$)
```

ステージ2 : オシロスコープをトーカに、コントローラとプリンタをリスナにして、データをSCRATCH.DATスクラッチ・ファイルに格納しながら印刷します。

```
CMD2$="?_D%":CALL IBCMD (BRD0%, CMD2$)
FILE$="SCRATCH.DAT":CALL IBRDF (BRD0%, FILE$)
```

3．オシロスコープがトーカになりデータを直接プリンタに送信

コントローラはスタンバイ・モードになり、データ印刷が完了すると、つまりEOIを検出するとGPIB動作に戻ります。

ステージ1 ： コントローラをトーカに、オシロスコープをリスナにして、スクリーン・ダンプ・コマンドを発行します。

```
CMD1$="?_@$":CALL IBCMD (BRD0%, CMD1$)
CMD$="SCDP":CALL IBWRT (BRD0%, CMD$)
```

ステージ2 ： オシロスコープをトーカに、プリンタをリスナにして、コントローラをスタンバイ・モードにします。

```
CMD2$="?_D%":CALL IBCMD (BRD0%, CMD2$)
V%=1:CALL IBGTS (BRD0%, V%)
```

注11: スキーマ2と3ではボード・レベルGPIBファンクション・コールを用いています。ここでは、コントローラ（ボード）、オシロスコープ、プリンタにはそれぞれアドレス0、4、5が割り当てられていると想定しています。コントローラ、オシロスコープ、プリンタに対するリスナ・アドレスとトーカ・アドレスを以下に示します。

論理デバイス	リスナ・アドレス	トーカ・アドレス
コントローラ	32（ASCII<space>	64（ASCII @）
オシロスコープ	32+4=36（ASCII $）	64+4=68（ASCII D）
ﾊｰﾄﾞｺﾋﾟｰ･ﾃﾞﾊﾞｲｽ	32+5=37（ASCII %）	64+5=69（ASCII E）

注12: コマンド文字列内の"?"と"_"はそれぞれアンリスン（unlisten）とアントーク（untalk）を表します。これらは、各デバイスを"既知（known）"ステートに設定するときに用います。

注13: プログラム例ではサイズを小さくするために、デバイスのトークとリスンの初期化命令が文字列内にグループ化されています。これらを以下に示します。

CMD1$="?_@$" アンリスン、アントーク、PCトーカ、DSOリスナ

CMD2$="?_D$" アンリスン、アントーク、PCリスナ、DSOトーカ

RS-232-Cで操作

LeCroyのオシロスコープは、RS-232-Cポートを介して、端末やコンピュータなどのホストよりリモート・コントロールできます。それには、"UTILITIES"メニューのGPIB/RS232サブメニューでオシロスコープをRS-232コントロールに設定する必要があります。

システム・コマンドのセクションに記述されているコマンドはすべてサポートされていますが、波形転送だけはヘキサ（HEX）モードでのみ転送できます。COMM_FORMATの値はデフォルトで適宜設定されています。WF?に対する応答構文はGPIBの場合と同じです。

この章では、RS-232-Cコマンド固有のものをいくつか定義します。これらは、オシロスコープを構成するとき、あるいはオシロスコープをリモート・モードやローカル・モードに設定するようなGPIB 488.1メッセージ類をシミュレートするときに用います。

表　記

このセクションでは、ASCIIで印字できない文字類はそれぞれのニーモニックで表わします。

例

<LF>　　10進数では10のASCII改行文字

<BS>　　10進数では8のASCIIバックスペース文字

CTRL_U　　コントロール・キーとUキーを同時に押すことを示します。

RS-232-Cピン配列

リモートRS-232-Cピン配列（後ろのパネルに表示されています）を以下に示します。

DB9 ピン番号		説明
3	TXD	転送データ（オシロスコープより）
2	RXD	受信データ（オシロスコープへ）
7	RTS	送信要求（Request To Send）（オシロスコープより）。ソフトウェアXon/Xoffハンドシェイクが選択されていると常にTRUE。それ以外（ハードウェア・ハンドシェイク）の場合、オシロスコープが文字受信可能時はTRUE、オシロスコープが文字受信不能時はFALSE。
8	CTS	送信クリア（Clear To Send）（オシロスコープへ）。TRUEのときオシロスコープ送信可能。FALSEのとき送信停止。オシロスコープがハードウェア・ハンドシェイクで出力するときに使用。
4	DTR	データ端末レディー（Data Terminal Ready）（オシロスコープより）。常にTRUE。
—	GND	保護接地
5	SIG GND	信号接地

RS-232-C構成

RS-232-Cポートは全二重で構成されます。これは、コントローラとオシロスコープの両方で、同時にメッセージ送受信可能であることを意味します。しかし、オシロスコープは新しいコマンドを受信するときには、出力を停止します。

オシロスコープへの長いメッセージの転送は、トリガ・モードで捕捉処理中でにない時に実行する必要があります。これは、波形、あるいはオシロスコープのフロント・パネル・セットアップを送信する時に特に重要です。

RS-232-Cポートの挙動は、オペレータの必要性に応じて設定できます。そのために、フロント・パネルの基本セットアップの他に、"即時コマンド（immediate command）"と"COMM_RS232"特殊コマンドがあります。即時コマンドは、ASCII エスケープ（ESCape）文字<ESC>（10進数では27）の後に他の文字を続けたもので構成されます。このコマンドは、2番目の文字が受信されると即座に解読されます。

注： RS-232-Cのボー・レート、パリティ、文字長、ストップ・ビット数は、フロント・パネルの"SAVE"ボタンや"RECALL"ボタンで、あるいは"*SAV"、"RCL"、または"PANEL_SETUP"で保存された、あるいはリコールされたパラメータの中にあります。リコール時は、これらのパラメータが実際のものと同じ値に設定されているか確認する必要があります。この値が異なると、ホストはオシロスコープと通信できなくなることがあり、この場合手動で再構成しなければなりません。

受信文字のエコー

シリアル・ポートでは受信文字をエコーできます。オシロスコープを端末に接続しているときは、エコーは有用です。エコーのオン/オフは、それぞれ<ESC>]または<ESC>[の2文字シーケンスを送信して切り替えることができます。デフォルトはオンです。

注： ホストは、オシロスコープからの受信文字をエコーできません。

ハンドシェイク・コントロール

オシロスコープは入力バッファが満杯に近づくと、ホストにハンドシェイク信号を送信して、転送を中断するよう指示します。逆に、このバッファに十分空きができて、さらに文字を受信できるようになると、もう1つのハンドシェイク信号を送信します。ハンドシェイク信号は、CTRL_S（または<XOFF>）文字とCTRL_Q（または<XON>）文字、またはRTSライン上の信号レベルです。これは、XON/XOFFハンドシェイク（これがデフォルトです）には2文字シーケンス<ESC>]の送信で、RTSハンドシェイクには<ESC>[で選択されます。

オシロスコープより送信される文字フローは、CTSライン上の信号レベルで、または<XON>/<XOFF>の文字ペアでコントロールできます。

編集機能

オシロスコープを端末に直接接続しているときは、下記の機能を用いてタイプ・エラーを訂正できます。

<BS>または<DELETE>	最終文字を削除
CTRL_U	最終行を削除

メッセージ終端文字

"メッセージ終端文字（message terminator）"は、受信側にメッセージ送信が完了したことを知らせるマーカです。

オシロスコープへの入力では、プログラム・メッセージ終端文字（Program Message Terminator）はユーザ選択可能な1文字です。他の用途には絶対使われないような文字を選択します。この文字は、COMM_RS232コマンドとキーワードEIで選択します。デフォルトのプログラム・メッセージ終端文字はASCII文字の<CR>、10進数では13です。

オシロスコープは、応答のたびに応答文字の末尾に応答メッセージ終端文字（Response Message Terminator）を付けます。これは、コンピュータ・プロンプトのようなユーザ選択可能な文字列です。この文字列は空にはできません。デフォルトの応答メッセージ終端文字は"\n\r"で、<LF><CR>を意味します。

例

(1) COMM_RS232 EI,3

このコマンドは、オシロスコープに、受信メッセージはそれぞれASCII文字の<ETX>（10進数の3に対応します）で終端されることを伝えます。

(2) COMM_RS232 EO,"\r\nEND\r\n"

このコマンドは、オシロスコープに、応答する文字列にはそれぞれ文字列"\r\nEND\r\n"を付けるように指示します。

上記設定後のホスト・コマンドは以下のようになります。

```
TDIV?<ETX>
```

オシロスコープの応答は以下のようになります。

```
TDIV 1.S
END
```

注： ホストは、COMM_RS232コマンドを送信したときは、オシロスコープがそれに対応した挙動に切り替わるのを待ってから、新しいモードでコマンドを送信する必要があります。そのための安全策として、COMM_RS232コマンド行にクエリを入れ、その応答が得られるまで待機します。以下に例を示します。

```
COMM_RS232 EI, 3; *STB?
```

SRQメッセージ

STatus Byte（STB）のMaster Summary Status（MSS）ビットが設定されるたびに、SRQメッセージ（1文字列）がホストに送られ、オシロスコープからのサービスリクエストが知らされます。RS-232-C SRQメッセージはGPIB SRQメッセージと同じ意味です。この文字列が空の場合、メッセージは送信されません。これはデフォルト設定です。SRQメッセージの末尾には応答メッセージ終端文字は付けられないので注意してください。

例

COMM_RS232 SRQ, "\r\n\nSRQ\r\n\a"

MSSビットが設定されると、オシロスコープは下記のように送信します。

1つの<CR>の後に2つの<LF>

SRQ

1つの<CR>の後に1つの<LF>

そしてブザーが鳴ります。

長い行の分割

行分割は、ある文字数以上の行を受け付けることができないホストのために準備された機能です。オシロスコープは、応答を複数行に分割して返すように構成できます。この機能は、どの応答メッセージにも適用できますが、波形転送やフロント・パネル・セットアップ情報の転送に有用です。この機能は2つのパラメータでコントロールします。

行区切り文字: オフ（Off）	このメッセージでは行は分割されません。
<CR>、<LF>、または<CR><LF>	行区切り文字にできます。
行の長さ	1行の最大文字数です。

例

COMM_RS232 LS, LF, LL, 40

行区切り文字はASCII文字の<LF>であり、1行の最大文字数は（区切り文字は除き）40文字です。

オシロスコープはPNSU?コマンドを受信すると、たとえば次のように応答します。

PNSU#9000001496

AAAA5555000655AA4030005800190000000000001

0000000000000000000000000000C1B0100580000

00000000000000000000000000000000000000000

…

備　考

オシロスコープに送信された長いコマンドは複数行に分割されません。以前のクエリの応答をオシロスコープへの送信コマンドに用いるときは、行区切り文字（<LF>および/または<CR>）は取り除く必要があります。

これは、オシロスコープに送信する文字列内の行区切り文字にもあてはまります。

しかし、オシロスコープに送信するヘキサASCIIデータには行区切り文字を入れることができます。行区切り文字を使うときは、データ内に入力メッセージ終端文字や行区切り文字が入っていないことを確認してください。

GPIBコマンドをシミュレートするコマンド

デバイス・クリア・コマンド<ESC>Cまたは<ESC>c

このコマンドは、入力バッファと出力バッファをクリアします。これは、GPIBのDCLメッセージまたはSDCメッセージと同じ意味を持ちます。

リモートに設定する（Set to Remote）コマンド<ESC>Rまたは<ESC>r（REN）

このコマンドは、オシロスコープをリモート・モードに設定します。この機能は、GPIBでRENラインをTRUEにしてオシロスコープをリモート・モードに設定するのと同じです。

ローカルに設定する（Set to Local）コマンド<ESC>Lまたは<ESC>l

このコマンドは、オシロスコープをローカル・モードに設定します。これはローカル・ロックアウトをクリアします。この機能は、GPIBでRENラインをFALSEにするのと同じです。

ローカル・ロックアウトに設定する（Set to Local Lockout）コマンド<ESC>Fまたは<ESC>f

このコマンドは、オシロスコープが既にリモート・モードのときは直ちに、そうでない場合はオシロスコープが次にリモート・コントロールに設定されたときに、フロント・パネルの"LOCAL"ボタンをディスエーブルにします。この"LOCAL"メニュー・ボタンのディスエーブルは、"ローカル・ロックアウト（Local Lockout）"と呼ばれ、<ESC>Lコマンドでのみキャンセルできます。<ESC>Fは、GPIBのLLOインターフェイス・メッセージと同じ意味を持ちます。

トリガ・コマンド<ESC>Tまたは<ESC>t（GET）

このコマンドは、オシロスコープが"STOP"モードのとき再びトリガ・システムにします（オシロスコープがリモート・モードのときにのみ有効です）。これは、"*TRG"コマンド、およびGPIBのGETインターフェイス・メッセージと同じ意味を持ちます。

波形構造

波形について

本章では、波形データのリード/ライトと、その内容について説明します。

波形データは基本的に次の2つに分けることができます。捕捉によってADCより得られた生のデータである基本データ配列と、それに付随する記述情報です。この記述情報には、垂直スケール、水平スケール、日時などデータ解釈になくてはならないものが含まれます。

波形情報には、INSPECT?クエリでアクセスできます。これにより、波形情報は簡単に理解できるASCIIテキスト形式に変換されます。また、WAVEFORM?クエリでより高速に転送することも、WAVEFORMコマンドでオシロスコープに書き戻すこともできます。オシロスコープには、波形情報の編成方法を詳細に記述するテンプレートと呼ばれるデータ構造があります。

代わりに、STOREコマンドとSTORE_SETUPコマンドで、波形データをよく知られているスプレッドシートや数理処理パッケージなどの、あらかじめフォーマットされたASCII出力形式[1]に格納することもできます。

波形の論理データ・ブロック

テンプレートによって、波形の論理データ・ブロックの形式と内容に関する詳細な記述が得られます。テンプレートは参照用に提供されます。TEMPLATE?クエリおよびオシロスコープで用いられる実際のテンプレートの使用が推奨されますが、付録Bにテンプレート例を示します。テンプレートは、オシロスコープのファームウエア拡張時に変更されることがあります。テンプレートは、波形解釈の後方互換性をとるときにも役立ちます。

通常、波形データは、波形記述子ブロック（1.）とデータ配列ブロック（5.）で構成されます。さらに複雑なケースでは、他のブロックが複数存在します。これらのデータ・ブロックを以下に示します。

1.波形記述子ブロック（WAVEDESC）

このブロックには、データより波形表示を再構成するときに必要な以下の情報が入っています。

－アクイジションのハードウェア設定情報
－厳密なイベント時刻
－実行済みのプロセス種類
－オシロスコープのモデル名と製造番号
－データ・ブロックに用いるコード化フォーマット
－その他常数

2.オプションのユーザ記述テキスト（USERTEXT）

WFTXコマンドで波形のタイトルや記述内容をこのブロックに書き込みます。逆にWFTX?クエリで読み出します。このテキスト・ブロックには160文字まで書き込めます。これらは、TEXT + TIMESステータス・メニューに1行40文字、4行で表示できます。

3.シーケンス捕捉時刻ブロック（TRIGTIME）

このブロックは、シーケンス捕捉でセグメントごとに厳密なタイミング情報を記録するときに使用されます。ここには、最初のセグメントのトリガを基準としたトリガごとの時刻と、そのトリガに関連するセグメントごとの最初のデータ・ポイント時刻が書き込まれます。

4.ランダム・インターリーブ・サンプリング時刻（RISTIME）

このブロックは、RIS捕捉でセグメントごとに厳密なタイミング情報を記録するときに使用されます。

5.データ配列ブロック（SIMPLEまたはDATA_ARRAY_1）

これは、波形の基本的な整数データです。これには、生のADCデータや修正したADCデータあるいは整数の波形処理結果が入れられます。

6.第2データ配列ブロック（DATA_ARRAY_2）

この第2データ配列は、極値関数（Extrema）（WP01オプション）やFFTの複素数（Complex）（WP02オプション）などの処理関数結果を保持するときに使用されます。この場合データ配列は下記の内容になります。

	Extrema	FFT
DATA_ARRAY_1	ルーフ・トレース	実数部
DATA_ARRAY_2	フロア・トレース	虚数部

注 ： TEMPLATEでは、DUALと名付けられた配列も記述されます。これは、INSPECT?コマンドで2つのデータ配列を一緒に調べられるようにするためです。

INSPECT?クエリ

これは、波形内容を調べる最も簡単な方法です。これは、データ部にも記述部にも使えます。最も簡単な形式を以下に示します。

```
INSPECT?  "name"
```

ここでは、テンプレートにより記述子項目名またはデータ・ブロック名が与えられます。回答は1文字列で返されますが、複数の行に広げることもできます。以下に一般的ダイアログをいくつか示します。

```
問い      C1:INSPECT? "VERTICAL_OFFSET"
応答      C1:INSP "VERTICAL_OFFSET :1.5625e-03"
問い      C1:INSPECT? "TRIGGER_TIME"
応答      C1:INSP "
          TRIGGER_TIME :Date=FEB 17, 1994, Time=4:4:29.5580
          "
```

INSPECT?は下記のように、波形記述子ブロック全体の読み出し可能変換を得ることもできます。

```
INSPECY? "WAVEDESC"
```

オシロスコープよりテンプレートをダンプすると、パラメータごとの解釈詳細が得られます。INSPECT?は、波形の測定データ値も調べられます。52ポイントの捕捉で下記のデータが得られます。

```
INSPECT? "SIMPLE"
C1:INSP "
  0.0005225  0.0006475   -0.00029    -0.000915   2.25001E-05  0.000835
  0.0001475  -0.0013525  -0.00204    -4E-05      0.0011475    0.0011475
  -0.000915  -0.00179    -0.0002275  0.0011475   0.001085     -0.00079
  -0.00179   -0.0002275  0.00071     0.00096     -0.0003525   -0.00104
  0.0002725  0.0007725   0.00071     -0.0003525  -0.00129     -0.0002275
  0.0005225  0.00046     -0.00104    -0.00154    0.0005225    0.0012725
  0.001335   -0.0009775  -0.001915   -0.000165   0.0012725    0.00096
  -0.000665  -0.001665   -0.0001025  0.0010225   0.00096      -0.0003525
  -0.000915  8.50001E-05  0.000835    0.0005225
  "
```

これらの数字は、ボルト単位の測定値に変換されます。もちろん、データ・ブロックに数千項目あるときは文字列は数多くの行にわたります。アプリケーションによっては、データ値ごとにBYTE（8ビット）やWORD（16ビット）で表した生の形式の方がよい場合もあります。この場合、WAVEFORM?と一緒に、測定値変換方法を指定する下記のパラメータを用いる必要があります。たとえば、次のように用います。

```
INSPECT?  "SIMPLE", BYTE
```

2つのデータ配列からなる波形のデータ値を調べるには、以下のように行います。

```
INSPECT? "DUAL"              1行でデータ値のペアが得られます。
INSPECT? "DATA_ARRAY_1"      最初のデータ配列の値が得られます。
INSPECT? "DATA_ARRAY_2"      2番目のデータ配列の値が得られます。
```

波形の一部を調べたり、間引き形式（sparsed form）で調べることもできます。これは、このセクションで後述しますがWAVEFORM_SETUPコマンドでコントロールします。

INSPECT?にはクエリ形式しかありません。これで、波形データをオシロスコープへ送り返すこ

とはできません。また、この方法で情報を送信することは冗長部分が多く、大変遅くなります。転送スピードやオシロスコープへ波形を戻す必要がある場合は、WAVEFORMコマンドを使用してください。

BASICユーザは、検査機能と波形クエリ・コマンドを組み合わせて、人が読める波形記述子のBASIC版と、オシロスコープへの再転送に適したフォーマットの全波形データとを共に含むファイルを作成できれば便利になると考えるかもしれません。これは、メモリにある波形に対して次のコマンドを送信して、その応答を直接ディスク・ファイルに返すことで実行できます。

```
MC:INSPECT? "WAVEDESC";WAVEFROM?
```

WAVEFORM?クエリ

WAVEFORMコマンドは、IEEE 488.2標準に定義されたブロック・フォーマットを用いて波形データを転送するには効果的な方法です。波形の論理ブロックはすべて次の1つのクエリで読み出すことができます。

```
C1:WAVEFORM?
```

これは、これだけで完結しており、WAVEFORMコマンドでその応答をそのままオシロスコープにダウンロードできるため、ほとんどのアプリケーションに推奨される形式です。次のようなクエリでは、任意のブロックを選択して読み出すことができます。

```
C1:WAVEFORM? DAT1
```

これは、同じ捕捉条件を持った波形を多数読み出すときに、あるいは大量の生の整数データのみ必要なときに、時間とスペースを節約できます。ブロック名詳細は、システム・コマンドのセクションのWAVEFORMを参照してください。

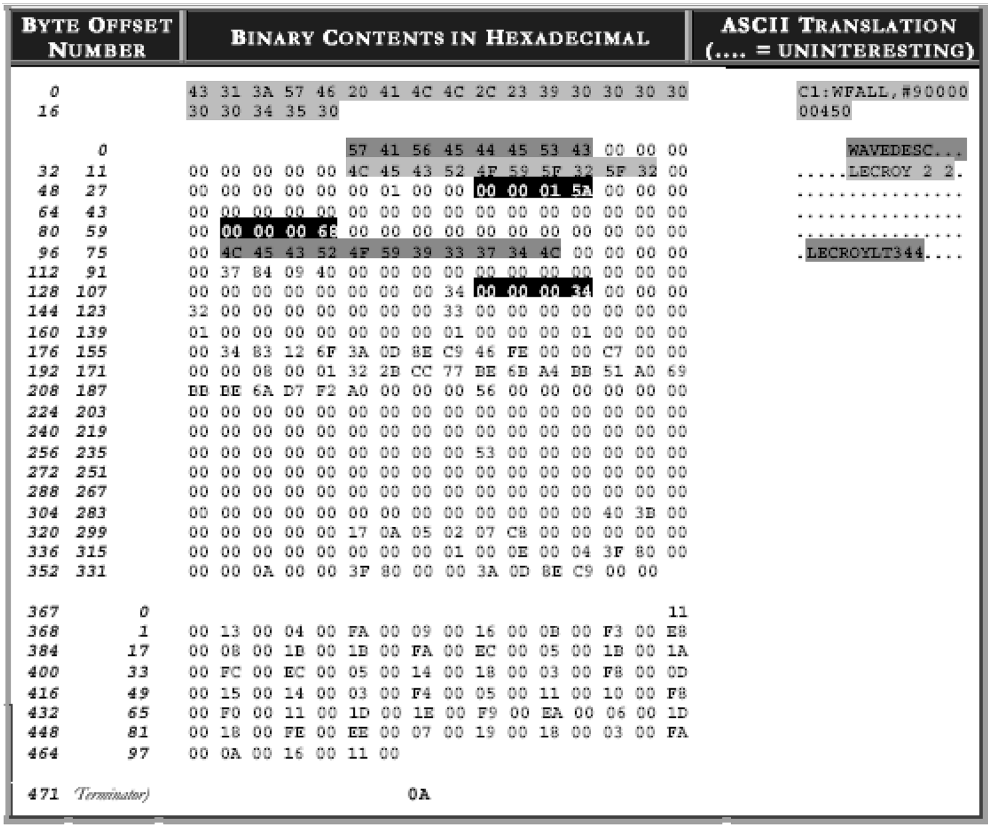
注 : 波形クエリの応答は、すぐにバイナリ・フォーマットで1600万バイトを超える、HEXオプションを使えばさらにその倍の大きいブロックになります。

波形記述子の変換

```
C1:WAVEFORM?またはC1:WAVEFORM? ALL
```

上記の形式のクエリ・コマンドに対するバイナリ応答をディスク・ファイルに入れて、下記の16進数形式とASCII形式にダンプすることができます。（これは、GPIBのデフォルト設定で実行されます）。

バイト・オフセット	16 進数形式のバイナリ内容	ASCII 変換 (.=不要部)



BYTE OFFSET NUMBER		BINARY CONTENTS IN HEXADECIMAL	ASCII TRANSLATION (.... = UNINTERESTING)
0		43 31 3A 57 46 20 41 4C 4C 2C 23 39 30 30 30 30	C1:WFALL,#90000
16		30 30 34 35 30	00450
0		57 41 56 45 44 45 53 43 00 00 00	WAVEDESC...
32	11	00 00 00 00 00 4C 45 43 52 4F 59 5F 32 5F 32 00	.....LECROY 2 2.
48	27	00 00 00 00 00 00 01 00 00 00 01 5A 00 00 00	..............Z...
64	43	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
80	59	00 00 00 00 68 00 00 00 00 00 00 00 00 00 00 00	....h...........
96	75	00 4C 45 43 52 4F 59 39 33 37 34 4C 00 00 00 00	.LECROYLT344....
112	91	00 37 84 09 40 00 00 00 00 00 00 00 00 00 00 00	.7..@...........
128	107	00 00 00 00 00 00 00 00 34 00 00 00 34 00 00 00	........4...4...
144	123	32 00 00 00 00 00 00 00 33 00 00 00 00 00 00 00	2.......3.......
160	139	01 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00	....
176	155	00 34 83 12 6F 3A 0D 8E C9 46 FE 00 00 C7 00 00	.4..o:...F......
192	171	00 00 08 00 01 32 2B CC 77 BE 6B A4 BB 51 A0 69	.....2+.w.k..Q.i
208	187	BB BE 6A D7 F2 A0 00 00 00 56 00 00 00 00 00 00	..j......V......
224	203	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
240	219	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
256	235	00 00 00 00 00 00 00 00 00 00 53 00 00 00 00 00	..........S.....
272	251	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
288	267	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
304	283	00 00 00 00 00 00 00 00 00 00 00 00 00 40 3B 00	.............@;.
320	299	00 00 00 00 00 00 17 0A 05 02 07 C8 00 00 00 00	...........H....
336	315	00 00 00 00 00 00 00 00 00 01 00 0E 00 04 3F 80 00	..............?..
352	331	00 00 0A 00 00 3F 80 00 00 3A 0D 8E C9 00 00	.....?...:....
367	0		11
368	1	00 13 00 04 00 FA 00 09 00 16 00 0B 00 F3 00 E8	.....z.........s.
384	17	00 08 00 1B 00 1B 00 FA 00 EC 00 05 00 1B 00 1A	.....z.l.....;..
400	33	00 FC 00 EC 00 05 00 14 00 18 00 03 00 F8 00 0D	.|.l.....x....x..
416	49	00 15 00 14 00 03 00 F4 00 05 00 11 00 10 00 F8	.....t.....x....
432	65	00 F0 00 11 00 1D 00 1E 00 F9 00 EA 00 06 00 1D	.p.....>.y.j.....
448	81	00 18 00 FE 00 EE 00 07 00 19 00 18 00 03 00 FA	.....~.n.....x..z
464	97	00 0A 00 16 00 11 00	..........q.....
471	(Terminator)	0A	

先頭の10バイトがASCIIに変換され、クエリ応答の開始が簡単にわかります。この後には文字列"#9000000450"が続き、これは、9個のASCII整数を用いて（450バイトの）ブロック長を指定するバイナリ・ブロックの先頭です。波形データそのものは、この直後のバイト番号21（先頭バイトがバイト番号0です）より始まっています。

波形記述子はテンプレート（付録Bを参照してください）で解読できます。これによると、最初のオブジェクトはDESCRIPTOR_NAMEであり、見えている値WAVEDESCを持つ16文字の文字列です。記述子の先頭より16バイト目（またはバイト番号37）では、次の文字列、つまり値LECROY_2_2を持ったTEMPLATE_NAMEであることがわかります。その後他のパラメータがいくつか続きます。記述子の先頭より76バイト目（またはバイト番号97）にINSTRUMENT_NAMEがあることが容易にわかります。

同様にして、記述子の長さを示す4バイト長の整数が36バイト目（またはバイト番号57）より始まっていることがわかります。

WAVE_DESCRIPTOR = 15A（ヘキサ） = 346

60バイト目（またはバイト番号81）では、他の4バイト整数でデータ配列の長さが示されているのがわかります。

WAVE_ARRAY_1 = 68（ヘキサ） = 104

116バイト目（またはバイト番号137）には、データ・ポイント数があります。

WAVE_ARRAY_COUNT = 34（ヘキサ） = 52

ここで、データは、記述子の先頭より346バイト目（またはバイト番号367）より始まり、52データ・ポイントのそれぞれが2バイトで表されていることがわかります。波形データの全長は346 + 104であり、この数字はこのブロックの先頭に示されているASCII文字列と同じです。バイト番号471の最後の0Aは、GPIBメッセージ終端文字<NL><EOL>に関連するNL文字です。

データは、バイト番号367より始まっていることがわかります。この例のデータはオシロスコープで8ビットADCで捕捉されているため、データ・ポイントはそれぞれ8ビットとそれに続く0バイトで構成されていることがわかります。他の多くの種類の波形では、この2バイト目は0にならず重要な情報が含まれます。データは、-32768 = 8000（ヘキサ）～32767 = 7FFF（ヘキサ）の値範囲を持つ符号付き形式でコード化されます。データ・フォーマットにBYTEオプションを選択すると、値は、-128 = 80（ヘキサ）～127 = 7F（ヘキサ）の値範囲の符号付き整数になります。これらのADC値は、以下のようにしてディスプレイ・グリッドにマップされます。

- グリッドの中心軸上が0に設定される。
- グリッドの最上部が127（BYTEフォーマット時）または32767（WORDフォーマット時）に設定される
- グリッドの最下部が-128（BYTEフォーマット時）または-32768（WORDフォーマット時）に設定される

波形垂直データの変換

この時点でデータの解読方法がわかったので、これを適当な測定値に変換できると便利です。

データ・ポイントごとの垂直測定値は、記述子より得られた垂直ゲインと垂直オフセットに依存します。捕捉波形では、これは、現在使用中のデータ表現に変換後選択された電圧軸（volts/div）と電圧オフセットに対応します。テンプレートにより、垂直ゲインとオフセットは、それぞれ記述子の156バイト目と160バイト目にあり、これらはIEEE 32ビット・フォーマットで浮動小数点数で保存されていることがわかります。垂直単位を示すASCII文字列は196バイト目のVERTUNITにあります。垂直値は次の関係式で与えられます。

value = VERTICAL_GAIN x データ - VERTICAL_OFFSET

上記データでは次のことがわかります。

177バイト目の浮動小数点数3483 126fより　　VERTICAL_GAIN = 2.4414e-07

181バイト目の浮動小数点数3A0D 8EC9より　　VERTICAL_OFFSET = 0.00054

217バイト目の文字列5600...より　　VERTICAL_UNIT = V = ボルト

したがって、

371バイト目の16進数ワードFA00より　　data[4] = FA00 = 64000

32767の最大16ビット値をオーバーフローしているため、負の値にする必要があります。2の補数変換を行うと、64000-2^{16}= -1536。

検査コマンドにあるように、　　value[4] = -0.000915 V

ご使用のコンピュータやソフトウェアがIEEE浮動小数点数を処理できないときは、付録Bのテンプレートにある、このフォーマットの記述を参照してください。

波形のデータ値はすべて測定値に対応するわけではありません。FIRST_VALID_PNTとLAST_VALID_PNTに必要な情報があります。WAVEFORM_SETUPコマンド・オプションを使用していると、記述子にSPARSING_FACTOR、FIRST_POINT、SEGMENT_INDEXも記録されるので、変換に役立ちます。

シーケンス捕捉では、セグメントごとのデータ値が正規の順番で与えられるので、セグメントは順番に読み出されます。重要な記述子パラメータはWAVE_ARRAY_COUNTとSUBARRAY_COUNTであり、これによりポイント総数とセグメント数が得られます。

extremaやFFT（複素数）などの波形では、得られる2つの結果に対する2つの配列が交互に与えられます。

データ・ポイントの水平位置の計算

どの垂直データ値にも、それぞれに対応する水平位置があり、通常は時間単位または周波数単位で測定されます。この位置計算は波形の種類に依存します。ここでは、単一スイープ波形、シーケンス波形、インターリーブ（RIS）波形を別々に取り扱います。どのデータ値にも位置iがあり、i = 0のオリジナル波形は捕捉データの第1ポイントに対応します。記述子パラメータHORUNITより水平単位名を持つ文字列が得られます。

単一スイープ波形

x[i]=HORIZ_INTERVAL x i + HORIZ_OFFSET.

捕捉波形では、この時間はトリガから対象データ・ポイントまでの時間です。HORIZ_OFFSETは各トリガについて測定されるため、これは捕捉ごとに異なります。

上記データでは次のようになります。

194バイト目の浮動小数点数322b cc77より　　HORIZ_INTERVAL = 1e-08
198バイト目の倍精度浮動小数点数be6b a4bb 51a0 69bbより HORIZ_OFFSET = -5.149e-08
262バイト目の文字列5300...より　　HORUNIT = S = 秒

したがって以下のようになります。

x[0]= -5.149e-08S
x[1]= -4.149e-08S

シーケンス波形

シーケンス波形は実に多くの独立した捕捉データで構成されるため、セグメントごとに固有の水平オフセットを持ちます。これらはTRIGTIME配列にあります。第nセグメントは以下のようになります。

x[i, n]=HORIZ_INTERVAL x i = TRIGGER_OFFSET[n].

TRIGTIME配列には、タイミング情報を、セグメントあたり2つの8バイト倍精度浮動小数点数で200セグメントまで入れることができます。

インターリーブ（RIS）波形

これらの波形は、互いにインターリーブされた数多くの捕捉データで構成されます。記述子パラメータRIS_SWEEPSより捕捉数が得られます。第iポイントが第mセグメントに属するとすると、

m=i modulo （RIS_SWEEPS）

であり、これは0～RIS_SWEEPS-1の値を持ちます。

したがって、

j = i - m

とすると、

x[i]=x[i,m]=HORIZ_INTERVAL x j + RIS_OFFSET[m]

ここで、各RIS_OFFSETはRISTIME配列にあります。このブロックには、8バイト倍精度浮動小数点数を100個まで入れることができます。オシロスコープは、下記式の時刻ごとにセグメントを得ようとします。

RIS_OFFSET[i]≅PIXEL_OFFSET + （i - 0.5） x HORIZ_INTERVAL.

したがって、仮にRISを、RIS_SWEEPS = 10、HORIZ_INTERVAL = 1 ns、PIXEL_OFFSET = 0.0とすると、ある特定イベントに対して次のようになります。

RIS_OFFSET[0]=-0.5ns RIS_OFFSET[1]=0.4ns
RIS_OFFSET[2]=1.6ns RIS_OFFSET[3]=2.6ns
RIS_OFFSET[4]=3.4ns RIS_OFFSET[5]=4.5ns

RIS_OFFSET[6]=5.6ns RIS_OFFSET[7]=6.4ns
RIS_OFFSET[8]=7.6ns RIS_OFFSET[9]=8.5ns

したがって、

x[0]=RIS_OFFSET[0]= -0.5ns
x[1]=RIS_OFFSET[1]= 0.4ns
...
x[9]=RIS_OFFSET[9]= 8.5ns
x[10]= 1ns x 10 + （-0.5）= 9.5ns
x[11]= 1ns x 10 + 0.4= 10.4ns
...
x[19]= 1ns' 10 + 8.5= 18.5ns
x[20]= 1ns' 20 + （-0.5）= 19.5ns
...

が得られます。

WAVEFORMコマンド

WAVEFORM?クエリで完全に読み込んだ波形は、そのままオシロスコープに戻すことができます。記述子には必要な情報がすべて入っているため、どの通信フォーマット・パラメータも気にする必要はありません。オシロスコープによって、必要な情報はすべて波形データより取り込まれます。

表示用にあるいは比較目的で波形を合成するときは、適当なサイズの波形を読み出して、そのデータを任意の値に置き換えます。これにより記述子のコヒーレンスが保証されます。

注 : 波形データはメモリ・トレース（M1、M2、M3、M4）にのみ戻せます。これは、WF?クエリの応答に含まれる接尾記号（C1やCHANNEL_1）は取り除いたり変更できることを意味します。第5章のWFコマンドの例に、この方法が紹介されています。

波形クエリをさらに制御

WAVEFORM?を使って作業を単純化しスピードを上げる方法が多数ありますが、代表的なものを以下にあげます。

波形の部分読み出し

WAVEFORM_SETUPコマンドにより、波形の短い部分の読み出しを設定できます。また、n個間隔でデータ・ポイントを読み出す間引き係数（sparsing factor）も選択できます。

バイト・スワップ

GPIBやRS-232-Cのリモート・コントロール・ポート経由でデータを送信するとき、COMM_ORDERコマンド（システム・コマンドのセクションを参照してください）を用いて、（記述子やデー

タ/時間配列に見られる）16ビット・ワード・フォーマットで表現されたデータの2バイトをスワップできます。これにより、ご使用のコンピュータ・システムに依存するデータ変換が容易になります。

- インテル・ベースのコンピュータには、データをLSBファーストで送信する必要があり、このコマンドはCORD LOとします。
- モトローラ・ベースのコンピュータには、データをMSBファーストで送信する必要があり、このコマンドはCORD HIですが、これは電源投入時のデフォルトです。

注： オシロスコープのオプションのメモリ・カード、ハードディスク、フロッピーディスクに書き込まれたデータは、常にLSBファースト（DOSデフォルト・フォーマット）です。CORDコマンドはこれらには使用できません。GPIBやRS-232-Cのリモート・コントロール・ポート経由でデータ送信するときにのみ使用できます。

データ長、ブロック・フォーマット、コード化

COMM_FORMATコマンドにより、これらのパラメータをコントロールできます。標準データ値の下位バイトの拡張精度が不必要な場合は、BYTEオプションにより、送信データや保存データの総量を2のべき乗の割合で削減できます。ご使用のコンピュータがバイナリ・データを読み込めないときは、HEXオプションにより、各バイト値を16進数のペアに変換して応答させることができます。

データのみの転送

COMM_HEADER OFFモードにより、WF? DAT1の応答をデータのみにできます（CI:WF DAT1部分の表示が消えます）。

COMM_FORMAT OFF,BYTE,BINを指定していると、その応答はデータ・バイトのみになります（#90000nnnnn部分の表示が消えます）。

RS-232-Cユーザ向けフォーマティング

COMM_RS232コマンドにより、非常に長いWF?応答を個別の行に分割できます。

高速波形転送

オシロスコープからコンピュータへの最大連続データ転送レートを達成するには、オペレータは数多くのファクタを最適化する必要があります。1つの最重要ポイントは、コンピュータの仕事量を限定することです。これは、データのディスク書き込みを避け、データ・ポイントあたりの計算量を最少化し、IOシステムへのコール数を最少化する、などを示します。オシロスコープについては、転送ポイント数を減らすか、ポイントあたりのデータ・バイト数を削減できます。パルス・パラメータ機能と処理関数を積極的に採用することによって、大幅な計算時間とデータ転送時間を削減できます。他の大変重要な原則を2点下記に示します。

- 波形転送でオーバーラップ波形捕捉を試みます。オシロスコープは、既に捕捉した波形や処理した波形を、新しい捕捉開始後に転送できます。オシロスコープがトリガを待つ必要のある場合（ライブ時間中）には、イベント獲得にかなりの時間をさくことができます。
- 転送ごとに多くのトリガが加算されるシーケンス・モードで波形転送数を最少化します。WAVEFORM_SETUPコマンドを使用して、転送するデータ・ポイント数を削減するよりも、この方法が推奨されます。これはまた、オシロスコープの転送オーバーヘッドも著しく削減します。以下に、いくつかコマンドを使用するときのタイプ例を示します。

  ARM;WAIT;C1:WF? で、イベントを待ち、データを転送し、新しい捕捉を開始します。この行は、波形読み出し後直ちに"ループ"するようプログラムできます。

ステータス・レジスタの使用

ステータス・レジスタの拡張セットにより、オペレータは、オシロスコープの内部処理ステータスをいつでも迅速に調べることができます。これらのレジスタとステータス・レポート・システムは、IEEE 488.2勧告に準拠して設計されています。

関連機能は共通ステータス・レジスタにグループ化されています。ステータス・バイト・レジスタ（STB: Status Byte Register）や標準イベント・ステータス・レジスタ（ESR: Standard Event Status Register）などいくつかのレジスタはIEEE 488.2標準に必須です。しかし、他のレジスタはデバイス仕様であり、これらにはコマンド・エラー・レジスタ（CMR: Command Error Register）や実行エラー・レジスタ（EXR: Execution Error Register）があります。IEEE 488.2必須ステータス・レジスタに関連するコマンドは、先頭にアスタリスク<*>がついています。

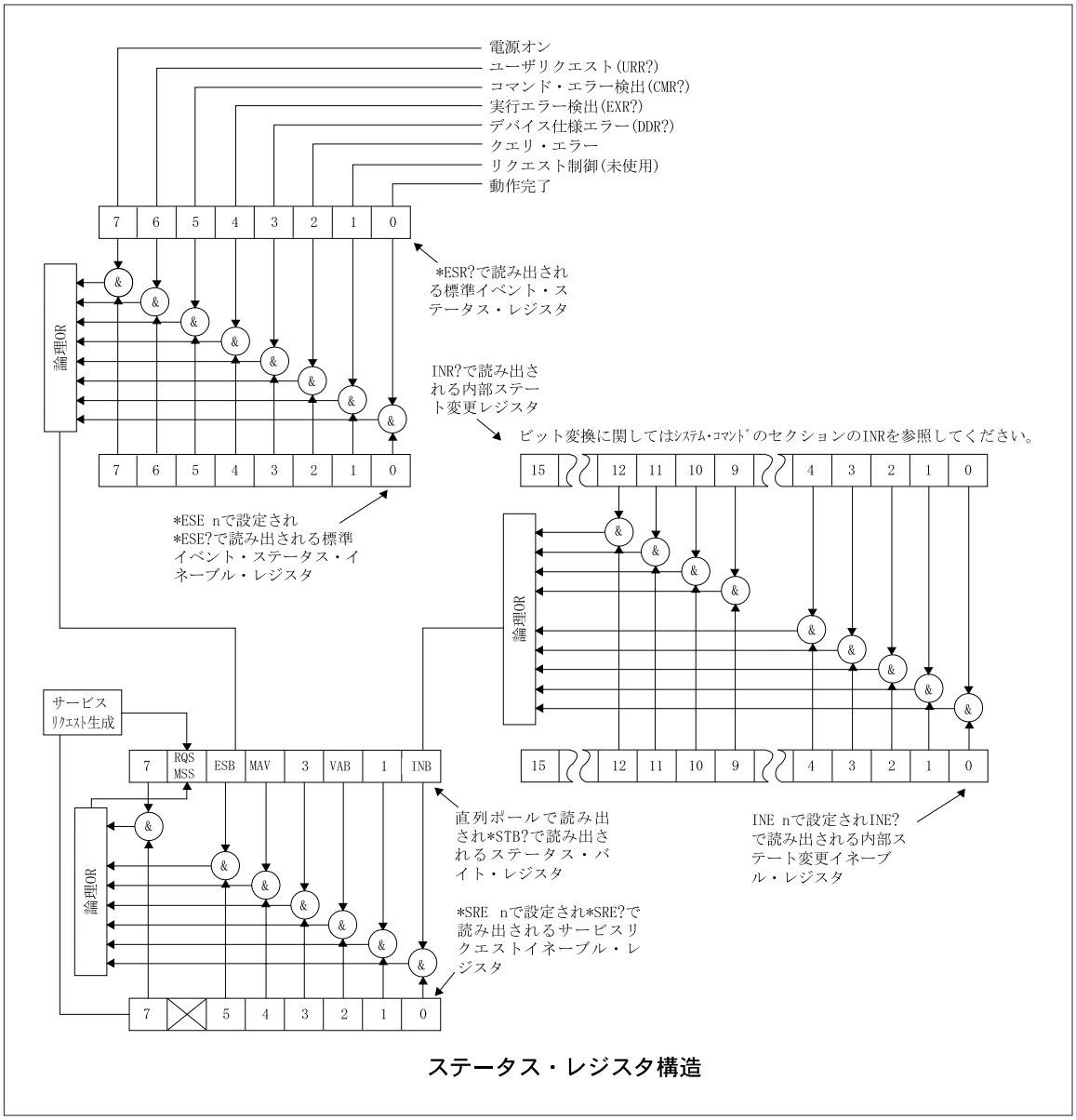
ステータスとサービスリクエストレポートの概要

ステータス・バイト・レジスタの標準イベント・ステータス・ビット（ESB: Standard Event Status Bit）と内部ステータス変更ビット（INB: Internal Status Change Bit）は、標準イベント・ステータス・レジスタ（ESR: Standard Event Status Register）と内部ステート変更レジスタ（INR: Internal State Change Register）のサマリ・ビットです。出力キューにデータ・バイトがあると常に、メッセージ可能ビット（MAV: Message Available Bit）が設定されます。値適用ビット（VAB: Value Adapted Bit）は、以前のコマンド解釈中にパラメータ値が適用されたことを示します。（たとえば、"TDIV 2.5 US"コマンドが受信されると、タイムベースが2μs/divに設定され、VABビットが設定されます）。

マスタ・サマリ・ステータス（MSS: Master Summary Status）ビットは、オシロスコープからのサービスリクエストがあることを示します。MSSビットは、サービスリクエストイネーブル・レジスタ（SRE: Service Request Enable Register）とともにSTBの他のビットのどれかがイネーブルのときにのみ、設定できます。

イネーブル・レジスタ（SRE、ESE、INE）はすべて、関連ステータス・レジスタとビット型のANDを生成するときに用いられます。この操作の論値ORはSTBレジスタにレポートされます。電源投入時、イネーブル・レジスタはすべて0であり、STBへのレポートはすべて禁止されます。

標準イベント・ステータス・レジスタ（ESR: Standard Event Status Register）はほとんどのエラーをサマライズし、内部ステート変更レジスタ（INR: Internal State Change Register）はオシロスコープの内部変更をレポートします。ESRにてレポートされるエラーの詳細は、"CMR?"、"DDR?"、"EXR?"、"URR?"の各クエリで取り込むことができます。



ステータス・レジスタ構造

レジスタ構造には、上図に表示されていないレジスタがもう1つあります。これはパラレルポール・イネーブル・レジスタ（PRE: Parallel Poll Enable Register）で、サービスリクエストイネーブル・レジスタ（SRE: Service Request Enable Register）と全く同じ動作をしますが、パラレルポールに用いられる"ist"ビット（これも図には表示されていません）を設定します。"ist"ビットは"*IST?"クエリで読み出すことができます。

例

たとえば"TRIG_MAKE SINGLE"など間違ったリモート・コマンドがオシロスコープに送信されると、そのコマンドは拒否され、コマンド・エラー・レジスタ（CMR: Command Error Register）が値1（判読不能コマンド/クエリ・ヘッダ）に設定されます。CMRの非0値は標準イベント・ステータス・レジスタ（ESR: Standard Event Status Register）のビット5にレポートされ、このビットが設定されます。

（"*ESE 32"コマンドで）標準イベント・ステータス・イネーブル・レジスタ（ESE: Standard Event Status Enable Register）のビット5が設定され、ステータス・バイト・レジスタ（STB: Status Byte Register）のESBサマリ・ビットへレポートするESRのビット5がイネーブルに設定されていない限り、なにも起こりません。

STBのESBサマリ・ビット設定がイネーブルのときは、（"*SRE 32"コマンドで）サービスリクエストイネーブル・レジスタの対応するビットが設定されてさらにレポーティングがイネーブルにされない限り、なにも起こりません。この場合、CMRの非0値の生成によってマスタ・サマリ・ステータス（MSS: Master Summary Status）ビットが設定され、サービスリクエスト（SRQ: Service Request）が生成されます。

CMRの値は、いつでも"CMR?"コマンドで読み出すと同時にリセットできます。コマンド・エラーの発生は、"ESR?"の応答を解析することでも検出できますが、いくつかのタイプのエラーが発生しているような場合は、イネーブル・レジスタESEとINEを用いて、注目しているエラーをSTBに伝搬させるほうが一般に効率的です。

サマリ

コマンド・エラー（CMR）によりESRのビット5が設定される場合を以下に示します。

— ESEのビット5が設定され、STBのESBも設定されたとき

— SREのビット5が設定され、STBのMSS/RQSも設定され、サービスリクエストが生成されたとき

ステータス・バイト・レジスタ（STB）

ステータス・バイト・レジスタ（STB: Status Byte Register）は、オシロスコープの中央レポーティング構造です。STBは1ビット・サマリ・メッセージ8個（内3個は未使用です）で構成され、オシロスコープに実装されている関連データ構造のカレント・ステータスを反映します。

ビット0は、内部ステート変更レジスタのINBサマリ・ビットです。このビットは、INRのビットがINEレジスタの対応するビットによってイネーブルされているとき、INRのビットのどれかが

設定されると設定されます。

ビット2は、値適用ビットで、以前のコマンド解釈中にパラメータ値が適用されたことを示します。

ビット4は、メッセージ可能（MAV: Message Available）ビットであり、インターフェイス出力キューが空でないことを示します。

STBの**ビット5**は、標準イベント・ステータス・レジスタのESBサマリ・ビットです。このビットは、ESRのビットがESEレジスタの対応するビットによってイネーブルされているとき、ESRのビットのどれかが設定されると設定されます。

ステータス・バイト・レジスタ（STB: Status Byte Register）の**ビット6**は、2つの方法で読み出せるため、マスタ・サマリ・ステータス（MSS: Master Summary Status）ビット、またはサービス・リクエスト（RQS: Request for Service）ビットと呼ばれます。"*STB?"コマンドはクエリ・モードでSTBを読み出しクリアしますが、この場合、STBのビット6はMSSビットであり、オシロスコープにサービス・リクエストの原因があるかどうかを示します。STBを読み出すもう1つの方法はシリアルポールです（第3章のGPIBシリアルポール手順を参照してください）。この場合、STBのビット6はRQSビットであり、オシロスコープがGPIBのSRQラインを起動していることを示します。シリアルポールはRQSビットのクリアのみ行います。したがって、STBのMSSビット（とMSS設定を引き起こした他のビットすべて）はシリアルポール後も設定されたまま保持されます。これらのビットはコントローラよりリセットする必要があります。

ステータス・バイト・レジスタは"STB?"クエリで読み出すことができます。応答には、レジスタのビット合計がバイナリ重みづけされて表示されます。このレジスタは、"STB?"、"ALST?"、"CLS"で、またはオシロスコープの電源投入時にクリアされます。

標準イベント・ステータス・レジスタ（ESR）

標準イベント・ステータス・レジスタ（ESR: Standard Event Status Register）は、イベント発生を示す16ビット・レジスタです。このレジスタのビット配列はIEEE 488.2で標準化されています。現在は下位8ビットのみが使用されています。

標準イベント・ステータス・レジスタは"*ESR?"クエリで読み出すことができます。応答には、レジスタのビット合計がバイナリ重みづけされて表示されます。このレジスタは、"*ESR?"クエリ、"ALST?"クエリ、"CLS"コマンドで、またはオシロスコープの電源投入時にクリアされます。

例

応答メッセージ"*ESR 160"は、コマンド・エラーが発生し、電源投入後初めてESRが読み出されていることを示します。値160は128（ビット7） + 32（ビット5）に分解できます。ビット設定に対応する状態に関しては、（システム・コマンドの）ESRコマンドの説明ページにある表を参照してください。

"電源オン（Power ON）"ビットは、電源投入後の最初の"ESR?"クエリでのみ表示されます。これは、このクエリによってこのレジスタがクリアされるためです。この種のコマンド・エラー

は、"CMR?"クエリでコマンド・エラー・ステータス・レジスタを読み出すことで判断できます。次のコマンド・エラーでESRのCMRビットを設定するには、このレジスタを必ずしも読み出す必要は（同時にクリアする必要も）ありません。

標準イベント・ステータス・イネーブル・レジスタ（ESE）

標準イベント・ステータス・イネーブル・レジスタ（ESE: Standard Event Status Enable Register）により、標準イベント・ステータス・レジスタの複数のイベントをSTBのESBサマリ・ビットにレポートすることができます。標準イベント・ステータス・イネーブル・レジスタは"*ESE"コマンドで変更できます。このレジスタは、"*ESE 0"コマンドで、または電源投入後にクリアされます。"*ESE?"クエリで読み出すことができます。

例

"*ESE 4"で、標準イベント・ステータス・イネーブル・レジスタのビット2（バイナリ4）が設定され、クエリ・エラーのレポートがイネーブルされます。

サービス・リクエスト・イネーブル・レジスタ（SRE）

サービス・リクエスト・イネーブル・レジスタ（SRE: Service Request Enable Register）は、ステータス・バイト・レジスタのどのサマリ・ビットでサービス・リクエストが生成されるようにするかを指定します。サービス・リクエスト・イネーブル・レジスタは8ビットで構成されます。このレジスタのビットのどれかを設定すると、関連するイベントがTRUEのときに、ステータス・バイト・レジスタの同じビット位置のサマリ・ビットでサービス・リクエストを生成できます。ビット6（MSS）は、設定できませんし、"*SRE?"クエリに対する応答では常に0とレポートされます。

標準イベント・イネーブル・レジスタは"*SRE"コマンドで変更されます。これは、"*SRE 0"コマンドで、または電源投入時にクリアされます。"*SRE?"クエリで読み出すことができます。

パラレルポール・イネーブル・レジスタ（PRE）

パラレルポール・イネーブル・レジスタ（PRE: Parallel Poll Enable Register）は、ステータス・バイト・レジスタのどのサマリ・ビットで"ist"個別ローカル・メッセージが設定されるようにするかを指定します。このレジスタは、サービス・リクエスト・イネーブル・レジスタによく似ていますが、MSSではなくパラレルポール"ist"の設定に用いられます。

"ist"の値は、パラレルポールなしで"*IST?"クエリでも読み出すことができます。その応答には、"ist"メッセージが設定されているかいないか（値が1か0か）が示されます。

パラレル・ポール・イネーブル・レジスタは"*PRE"コマンドで変更できます。これは、"*PRE 0"コマンドで、または電源投入時にクリアされます。"*PRE?"クエリで読み出すことができます。（第3章のGPIB並列ポール手順を参照してください）。

例

"*PRE 5"で、パラレル・ポール・イネーブル・レジスタのビット2と0（10進数の4と1）が設定されます。

内部ステート変更ステータス・レジスタ（INR）

内部ステート変更ステータス・レジスタ（INR: Internal State Change Status Register）は、数多くの内部動作の完了状態をレポートします。この16ビット型レジスタに記録されたイベントは、システム・コマンドのセクションにある"INR?"コマンドでリスンされます。

内部ステート変更ステータス・レジスタは"INR?"クエリで読み出すことができます。この応答は、レジスタのビット合計がバイナリ重み付けされて表示されます。このレジスタは、"INR?"クエリ、"ALST"クエリ、"*CLS"コマンドで、または電源投入時にクリアされます。

内部ステート変更イネーブル・レジスタ（INE）

内部ステート変更イネーブル・レジスタ（INE: Internal State Change Enable Register）は、内部ステート変更ステータス・レジスタの1以上のイベントをSTBのINBサマリ・ビットにレポートできるようにします。

内部ステート変更イネーブル・レジスタは"INE"コマンドで変更できます。このレジスタは、"INE 0"コマンドで、または電源投入時にクリアされます。"INE?"クエリで読み出すことができます。

コマンド・エラー・ステータス・レジスタ（CMR）

コマンド・エラー・ステータス・レジスタ（CMR: Command Error Status Register）には、オシロスコープで検出された最後のコマンド・エラーのコードが入っています。コマンド・エラーは"CMR?"コマンドでリストされます。

コマンド・エラー・ステータス・レジスタは"CMR?"クエリで読み出すことができます。その応答はエラー・コードです。このレジスタは、"CMR?"クエリ、"ALST"クエリ、"*CLS"コマンドで、または電源投入時にクリアされます。

デバイス依存エラー・ステータス・レジスタ（DDR）

デバイス依存エラー・ステータス・レジスタ（DDR: Device Dependent Error Status Register）は、オシロスコープに影響を与えるハードウェア・エラーの種類を示します。このレジスタの個々のビットで特定のハードウェア障害がレポートされます。これらは"DDR?"コマンドでリスンされます。

デバイス依存エラー・ステータス・レジスタは"DDR?"クエリで読み出すことができます。この応答は、レジスタのビット合計がバイナリ重み付けされて表示されます。このレジスタは、"DDR?"クエリ、"ALST"クエリ、"*CLS"コマンドで、または電源投入時にクリアされます。

実行エラー・ステータス・レジスタ（EXR）

実行エラー・ステータス・レジスタ（EXR: Execution Error Status Register）には、オシロスコープで検出された最後の実行エラーのコードが入っています。実行エラー・コードは"EXR?"コマンドでリストされます。

実行エラー・ステータス・レジスタは"EXR?"クエリで読み出すことができます。その応答はエラー・コードです。このレジスタは、"EXR?"クエリ、"ALST"クエリ、"*CLS"コマンドで、または電源投入時にクリアされます。

ユーザ・リクエスト・ステータス・レジスタ（URR）

ユーザリクエストステータス・レジスタ（URR: User Request Status Register）には、最後に押されたメニュー・ボタンのIDコードが入っています。これらのコードは"URR?"コマンドでリストされます。

ユーザリクエストステータス・レジスタは"URR?"クエリで読み出すことができます。その応答は、選択されたメニュー・ボタンに関連する10進数コードです。このレジスタは、"URR?"クエリ、"ALST"クエリ、"*CLS"コマンドで、または電源投入時にクリアされます。

GPIBプログラム例

例1 対話式GPIBプログラム'IBIC'の使用

この例では、ナショナル・インスツルメンツのGPIBインターフェイス・カードを装備したIBM PCまたはその互換コンピュータを想定しています。また、GPIBドライバはデフォルト・ステートのままであり、デバイス名"dev4"は、オシロスコープのアドレスと見なされるGPIBアドレス4に対応すると想定しています。ユーザが入力するテキストにはすべて下線が付けられています。

```
IBIC<cr>
    プログラム見出し部分
:ibfind<CR>
    ボード/デバイス名を入力:dev4<CR>
dev4:ibwrt<CR>
    文字列を入力:"tdiv?"<CR>
[0100]          (cmpl)
count:5
dev4:ibrd<CR>
    バイト数を入力:10<CR>
[0100]          (cmpl)
count:10
54 44 49 56 20 35 30 45         TDIV  50E
2D 39                           -9
dev4:ibwrt<CR>
    文字列を入力:"c1:cpl?"<CR>
[0100]          (cmpl)
count:7
dev4:ibrd<CR>
    バイト数を入力:20<CR>
[2100]          (end cmpl)
カウント:11
43 31 3A 43 50 4C 20 44         C1:CPLD
35 30 0A                        50z
dev:4:q<CR>
    プログラム終了
```

例2 IBM PC用GPIBプログラム（高レベルファンクション・コール）

下記のBASICAプログラムにより、IBM PCをGPIBコントローラとして用いてオシロスコープを完全に対話式に制御できます。ここでも、コントローラにはナショナル・インスツルメンツのGPIBインターフェイス・カードが装備されていると想定しています。たとえば"c1:vdiv 50 mv"（引用符なしで）とコマンドのテキスト文字列を単純に入力して、システム・コマンドのセクションにリストされているリモート・コントロール・コマンドがすべて使えます。クエリの応答でオシロスコープより返された情報は、プログラムにより自動的に表示されます。

さらに、使いやすいようにいくつかのユーティリティも用意されています。STコマンドやRCコマンドで正しいドライブ名とファイル名を指定することにより、波形データをディスクに保存したりディスクから取り出したりできます。LCコマンドはオシロスコープをローカル・モードに戻します。オシロスコープより返される応答は文字列に変換されますが、255文字までに制限されています。

注1 : ナショナル・インスツルメンツのGPIBドライバGPIB.COMはデフォルト・ステートであると想定しています。これは、インターフェイス・ボードがそのシンボリック名'GPIB0'で参照でき、アドレス1～16のGPIB上のデバイスがそれぞれシンボリック名'DEV1'～'DEV16'でコールできることを意味しています。

注2 : 1～99行目はナショナル・インスツルメンツが提供するDECL.BASファイルのコピーです。先頭の6行はGPIBハンドラのイニシャライズに必要です。DECL.BASは、GPIBイニシャライズ時BIB.Mファイルにアクセスする必要があります。BIB.Mは、ナショナル・インスツルメンツが提供するファイルの1つですが、カレント・ディレクトリに入れておく必要があります。

注3 : DECL.BASの先頭の2行には各行に"XXXXX"文字列がありますが、これは、（BASICAで現在使用可能な空間よりBIB.Mサイズを差し引いて計算される）BASICAの最大ワークスペースを決定するバイト数で置き換える必要があります。たとえば、BIB.Mサイズが1200バイトで、BASICAロード時"60200 bytes free（バイト空き）"と表示された場合は、"XXXXX"の所に5900以下の値を入れます。

注4 : デフォルト・タイムアウト10秒は、このプログラム実行時は300 msに変更されます。しかし、GPIBハンドラのデフォルト値は変更されません。ユーザがリモート・コントロールに設定すると、プログラムはオシロスコープにIBWRTファンクション・コールでその設定を送信します。その後、オシロスコープからの応答の有無にかかわらず、プログラムは常にIBRDコールを実行します。応答は受信されると即座に表示されます。応答がない場合、プログラムはタイムアウトを待ってから次のコマンドを要請します。

例２

```
1-99    <DECL.BAS>
100     CLS
110     PRINT "Control of the 9300 via GPIB and IBM PC"
115     PRINT ""
120     PRINT "Options :     EX to exit LC local mode"
125     PRINT "    ST store data        RC recall data"
130     PRINT ""
140     LINE INPUT "GPIB-address of oscilloscope (1...16)? :",ADDR$
145     DEV$ = "DEV" + ADDR$
150     CALL IBFIND(DEV$,SCOPE%)
155     IF SCOPE% < 0 THEN GOTO 830
160     TMO% = 10  'timeout = 300 msec (rather than default 10 sec)
165     CALL    IBTMO(SCOPE%,TMO%)
170     '
200     LOOP% = 1
205     WHILE LOOP%
210             LINE INPUT "Enter command (EX --> Exit) : ",CMD$
220             IF CMD$ = "ex" OR CMD$ = "EX" THEN LOOP% = 0 : GOTO 310
230             IF CMD$ = "st" OR CMD$ = "ST" THEN GOSUB 600 : GOTO 300
240             IF CMD$ = "rc" OR CMD$ = "RC" THEN GOSUB 700 : GOTO 300
250             IF CMD$ = "lc" OR CMD$ = "LC" THEN GOSUB 400 : GOTO 300
260             IF CMD$ = "" THEN GOTO 300
270             CALL IBWRT(SCOPE%,CMD$)
275             IF IBSTA% < 0 THEN GOTO 840
280             GOSUB 500
300     WEND
310     GOSUB 400
320     END
400     '
405     'SUBROUTINE LOCAL_MODE
```

```
410     '
420     CALL IBLOC(SCOPE%)
425     PRINT ""
430     RETURN
500     '
505     'SUBROUTINE GET_DATA
510     'If there are no data to read, simply wait until timeout occurs
515     '
520     CALL IBRD(SCOPE%,RD$)
525     I = IBCNT% 'IBCNT% is the number of characters read
530     FOR J = 1 TO I
535               PRINT MID$(RD$,J,1);
540     NEXT J
545     PRINT ""
550     RETURN
600     '
605     'SUBROUTINE STORE_DATA
610     '
615     RD1$=SPACE$(3)
620     LINE INPUT "Specify trace (TA...TD,M1...M4,C1...C4): ",TRACE$
625     LINE INPUT "Enter filename : ",FILE$
630     CMD$="WFSU NP,0,SP,0,FP,0,SN,0; CHDR SHORT"
640     CALL IBWRT(SCOPE%,CMD$)
645     CMD$=TRACE$+":WF?"
650     CALL IBWRT(SCOPE%,CMD$)
660     CALL IBRD(SCOPE%,RD1$)      'Discard first 3 chars of response
665     CALL IBRDF(SCOPE%,FILE$)
670     IF IBSTA% < 0 THEN GOTO 840
675     PRINT ""
680     RETURN
700     '
705     'SUBROUTINE RECALL_DATA
```

```
710     '
715     LINE INPUT "Specify target memory (M1...M4):",MEM$
720     LINE INPUT "Enter filename : ",FILE$
730     CMD$=MEM$+":"
735     CALL IBWRT(SCOPE%,CMD$)
740     CALL IBWRTF(SCOPE%,FILE$)
745     IF IBSTA% < 0 THEN GOTO 840
750     PRINT ""
755     RETURN
800     '
810     'ERROR HANDLER
820     '
830     PRINT "IBFIND ERROR"
835     END
840     PRINT "GPIB ERROR -- IBERR: ";IBERR%;"IBSTA: ";HEX$(IBSTA%)
END
```

例3 IBM PC用GPIBプログラム（低レベルファンクション・コール）

この例には、例2と同じ関数がありますが、低レベル・ファンクション・コールで書かれています。

このプログラムでは、コントローラ（ボード）とオシロスコープ（デバイス）のアドレスがそれぞれ0と4であると想定しています。したがって、コントローラとデバイスのリスナとトーカのアドレスは以下のようになります。

	リスナ・アドレス	トーカ・アドレス
コントローラ	32 （ASCII <space>	64 （ASCII @）
デバイス	32+4=36（ASCII（ASCII $）	64+4=68 （ASCII D）

例３

```
1-99    <DECL.BAS>
100     CLS
110     PRINT "Control of the 9300 (address 4) via GPIB and IBM PC"
115     PRINT "": PRINT "Options : EX to exit     LC local mode"
120     PRINT "       ST store data   RC recall data": PRINT""
125     LOOP=1
130     CMD1$ = "?_@$" 'Unlisten, Untalk, Board talker, Device listener
135     CMD2$ = "?_ D" 'Unlisten, Untalk, Board listener, Device talker
140     BDNAME$= "GPIB0": CALL IBFIND(BDNAME$,BRD0%)
145     IF BRD0% < 0 THEN GOTO 420
150     CALL IBSIC(BRD0%): IF IBSTA% < 0 THEN GOTO 425
155     WHILE LOOP
160             LINE INPUT "Enter command (EX --> Exit) : ",CMD$
165             V% = 1: CALL IBSRE(BRD0%,V%)
170             IF CMD$ = "ex" OR CMD$ = "EX" THEN LOOP = FALSE: GOTO 205
175             IF CMD$ = "st" OR CMD$ = "ST" THEN GOSUB 285: GOTO 200
180             IF CMD$ = "rc" OR CMD$ = "RC" THEN GOSUB 365: GOTO 200
185             IF CMD$ = "lc" OR CMD$ = "LC" THEN  GOSUB 240: GOTO 200
190             IF CMD$ = "" THEN GOTO 200
195             CALL IBCMD(BRD0%,CMD1$): CALL IBWRT(BRD0%,CMD$): GOSUB 270
200     WEND
205     CALL IBSIC(BRD0%): V%=0: CALL IBSRE(BRD0%,V%)
210     CALL IBSIC(BRD0%)
215     END
220     '
230     'LOCAL MODE
235     '
240     V% = 0: CALL IBSRE(BRD0%,V%): PRINT ""
245     RETURN
250     '
```

```
260     'SUBROUTINE GET_DATA
265     '
270     CALL IBCMD(BRD0%,CMD2$): CALL IBRD(BRD0%,RD$): I=IBCNT%
275     FOR J=1 TO I: PRINT MID$(RD$,J,1);: NEXT J: PRINT ""
280     RETURN
285     '
290     'SUBROUTINE STORE_DATA
295     '
300     RD1$=SPACE$(3)
305     LINE INPUT "Specify trace (TA...TD,M1...M4,C1...C4):  ",TRACE$
310     LINE INPUT "Enter filename : ",FILE$
315     CALL IBCMD(BRD0%,CMD1$)
320     CMD$="WFSU NP,0,SP,0,FP,0,SN,0;CHDR SHORT"
321     CALL IBWRT(BRD0%,CMD$)
325     CMD$=TRACE$+":WF?": CALL IBWRT(BRD0%,CMD$)
330     CALL IBCMD(BRD0%,CMD2$): CALL IBRD(BRD0%,RD1$)
335     CALL IBRDF(BRD0%,FILE$)
340     IF IBSTA% < 0 THEN GOTO 430
345     PRINT ""
350     RETURN
355     '
360     'SUBROUTINE RECALL_DATA
365     '
370     LINE INPUT "Specify target memory (M1...M4): ",MEM$
375     LINE INPUT "Enter filename : ",FILE$
380     CALL IBCMD(BRD0%,CMD1$)
385     CMD$=MEM$+":": CALL IBWRT(BRD0%,CMD$)
390     CALL IBWRTF(BRD0%,FILE$)
395     IF IBSTA% < 0 THEN GOTO 430
400     PRINT ""
405     RETURN
410     '
```

```
415     'ERROR HANDLER
420     '
425     PRINT "IBFIND ERROR": STOP
430     PRINT "GPIB ERROR -- IBERR : ";IBERR%;"IBSTA : ";HEX$(IBSTA%)
435     STOP
440     END
```

波形テンプレート

"TMPL?"形式のコマンドに対するオシロスコープの応答を以下に示します。

```
/00
000000                  LECROY_2_3:  TEMPLATE
              8 66 111
;
; Explanation of the formats of waveforms and their descriptors on the
; LeCroy Digital Oscilloscopes,
;   Software Release 8.1.0, 98/09/29.
;
; A descriptor and/or a waveform consists of one or several logical data blocks
; whose formats are explained below.
; Usually, complete waveforms are read: at the minimum they consist of
;     the basic descriptor block WAVEDESC
;    a data array block.
; Some more complex waveforms, e.g. Extrema data or the results of a Fourier
; transform, may contain several data array blocks.
; When there are more blocks, they are in the following sequence:
;      the basic descriptor block WAVEDESC
;     the history text descriptor block USERTEXT (may or may not be present)
;    the time array block (for RIS and sequence acquisitions only)
;    data array block
;    auxiliary or second data array block
;
; In the following explanation, every element of a block is described by a
; single line in the form
;
; <byte position>   <variable name>: <variable type> ; <comment>
;
;  where
;
;   <byte position> = position in bytes (decimal offset) of the variable,
;              relative to the beginning of the block.
;
;   <variable name> = name of the variable.
;
;   <variable type> = string     up to 16-character name
;                          terminated with a null byte
;               byte      08-bit signed data value
```

```
;             word      16-bit signed data value
;             long      32-bit signed data value
;            float      32-bit IEEE floating point value
;                                                                  with the
format shown below
;                    31  30 .. 23  22 ... 0   bit position
;                     s  exponent   fraction
;                     where
;                    s = sign of the fraction
;                     exponent = 8 bit exponent e
;                   fraction = 23 bit fraction f
;                   and the final value is
;                    (-1)**s * 2**(e-127) * 1.f
;          double      64-bit IEEE floating point value
;                      with the format shown below
;                  63  62 .. 52   51 ... 0   bit position
;                   s   exponent   fraction
;                   where
;                   s = sign of the fraction
;                    exponent = 11 bit exponent e
;                  fraction = 52 bit fraction f
;                  and the final value is
;                   (-1)**s * 2**(e-1023) * 1.f
;            enum      enumerated value in the range 0 to N
;                  represented as a 16-bit data value.
;                  The list of values follows immediately.
;                  The integer is preceded by an _.
;       time_stamp      double precision floating point number,
;                     for the number of seconds and some bytes
;                  for minutes, hours, days, months and year.
;
;                   double seconds    (0 to 59)
;                 byte   minutes    (0 to 59)
;                 byte   hours    (0 to 23)
;                 byte   days    (1 to 31)
;                 byte   months    (1 to 12)
;                 word   year     (0 to 16000)
```

```
;                    word    unused
;                   There are 16 bytes in a time field.
;             data    byte, word or float, depending on the
;                       read-out mode reflected by the WAVEDESC
;                      variable COMM_TYPE, modifiable via the
;                          remote command COMM_FORMAT.
;           text   arbitrary length text string
;                     (maximum 160)
;    unit_definition    a unit definition consists of a 48 character
;                   ASCII string terminated with a null byte
;                   for the unit name.
;
;================================================================================
;
WAVEDESC: BLOCK
;
; Explanation of the wave descriptor block WAVEDESC;
;
;
<  0>       DESCRIPTOR_NAME: string  ; the first 8 chars are always WAVEDESC
;
< 16>        TEMPLATE_NAME: string
;
< 32>        COMM_TYPE: enum           ; chosen by remote command COMM_FORMAT
        _0  byte
        _1  word
         endenum
;
< 34>        COMM_ORDER: enum
        _0   HIFIRST
        _1   LOFIRST
         endenum
;
;
; The following variables of this basic wave descriptor block specify
; the block lengths of all blocks of which the entire waveform (as it is
; currently being read) is composed. If a block length is zero, this
```

```
; block is (currently) not present.
;
; Blocks and arrays that are present will be found in the same order
; as their descriptions below.
;
;BLOCKS :
;
< 36>           WAVE_DESCRIPTOR: long    ; length in bytes of block WAVEDESC
< 40>          USER_TEXT: long      ; length in bytes of block USERTEXT
< 44>         RES_DESC1: long     ;
;
;ARRAYS :
;
< 48>           TRIGTIME_ARRAY: long    ; length in bytes of TRIGTIME array
;
< 52>          RIS_TIME_ARRAY: long     ; length in bytes of RIS_TIME array
;
< 56>         RES_ARRAY1: long      ; an expansion entry is reserved
;
< 60>          WAVE_ARRAY_1: long     ; length in bytes of 1st simple
                           ; data array. In transmitted waveform,
                           ; represent the number of transmitted
                            ; bytes in accordance with the NP
                               ; parameter of the WFSU remote command
                               ; and the used format (see COMM_TYPE).
;
< 64>          WAVE_ARRAY_2: long       ; length in bytes of 2nd simple
                         ; data array
;
< 68>           RES_ARRAY2: long
< 72>         RES_ARRAY3: long      ; 2 expansion entries are reserved
;
; The following variables identify the instrument
;
< 76>           INSTRUMENT_NAME: string
;
< 92>            INSTRUMENT_NUMBER: long
```

```
;
< 96>        TRACE_LABEL: string    ; identifies the waveform.
;
<112>          RESERVED1: word
<114>        RESERVED2: word        ; 2 expansion entries
;
; The following variables describe the waveform and the time at
; which the waveform was generated.
;
<116>          WAVE_ARRAY_COUNT: long   ; number of data points in the data
                        ; array. If there are two data
                          ; arrays (FFT or Extrema), this number
                        ; applies to each array separately.
;
<120>          PNTS_PER_SCREEN: long    ; nominal number of data points
                        ; on the screen
;
<124>          FIRST_VALID_PNT: long    ; count of number of points to skip
                        ; before first good point
                             ; FIRST_VALID_POINT = 0
                          ; for normal waveforms.
;
<128>          LAST_VALID_PNT: long     ; index of last good data point
                           ; in record before padding (blanking)
                          ; was started.
                                        ; LAST_VALID_POINT = WAVE_ARRAY_COUNT-1
                           ; except for aborted sequence
                           ; and rollmode acquisitions
;
<132>          FIRST_POINT: long        ; for input and output, indicates
                           ; the offset relative to the
                           ; beginning of the trace buffer.
                             ; Value is the same as the FP parameter
                               ; of the WFSU remote command.
;
<136>          SPARSING_FACTOR: long    ; for input and output, indicates
                           ; the sparsing into the transmitted
```

```
                    ; data block.
                      ; Value is the same as the SP parameter
                        ; of the WFSU remote command.
;
<140>          SEGMENT_INDEX: long      ; for input and output, indicates the
                    ; index of the transmitted segment.
                      ; Value is the same as the SN parameter
                        ; of the WFSU remote command.
;
<144>          SUBARRAY_COUNT: long     ; for Sequence, acquired segment count,
                        ; between 0 and NOM_SUBARRAY_COUNT
;
<148>          SWEEPS_PER_ACQ: long     ; for Average or Extrema,
                     ; number of sweeps accumulated
                   ; else 1
;
<152>          POINTS_PER_PAIR: word    ; for Peak Dectect waveforms (which always
                      ; include data points in DATA_ARRAY_1 and
                      ; min/max pairs in DATA_ARRAY_2).
                    ; Value is the number of data points for
                    ; each min/max pair.
;
<154>          PAIR_OFFSET: word        ; for Peak Dectect waveforms only
                    ; Value is the number of data points by
                    ; which the first min/max pair in
                      ; DATA_ARRAY_2 is offset relative to the
                      ; first data value in DATA_ARRAY_1.
;
<156>          VERTICAL_GAIN: float
;
<160>          VERTICAL_OFFSET: float   ; to get floating values from raw data :
                        ; VERTICAL_GAIN * data - VERTICAL_OFFSET
;
<164>          MAX_VALUE: float         ; maximum allowed value. It corresponds
                    ; to the upper edge of the grid.
;
<168>          MIN_VALUE: float         ; minimum allowed value. It corresponds
```

```
                        ; to the lower edge of the grid.
;
<172>          NOMINAL_BITS: word       ; a measure of the intrinsic precision
                        ; of the observation: ADC data is 8 bit
                        ;    averaged data is 10-12 bit, etc.
;
<174>          NOM_SUBARRAY_COUNT: word ; for Sequence, nominal segment count
                        ; else 1
;
<176>          HORIZ_INTERVAL: float    ; sampling interval for time domain
                        ;  waveforms
;
<180>          HORIZ_OFFSET: double     ; trigger offset for the first sweep of
                        ; the trigger, seconds between the
                        ; trigger and the first data point
;
<188>          PIXEL_OFFSET: double     ; needed to know how to display the
                        ; waveform
;
<196>          VERTUNIT: unit_definition  ; units of the vertical axis
;
<244>          HORUNIT: unit_definition   ; units of the horizontal axis
;
<292>          HORIZ_UNCERTAINTY: float ; uncertainty from one acquisition to the
                        ; next, of the horizontal offset in seconds
;
<296>          TRIGGER_TIME: time_stamp ; time of the trigger
;
<312>          ACQ_DURATION: float      ; duration of the acquisition (in sec)
                        ; in multi-trigger waveforms.
                        ; (e.g. sequence, RIS,  or averaging)
;
<316>             RECORD_TYPE: enum
            _0    single_sweep
            _1   interleaved
            _2    histogram
            _3    graph
```

```
          _4    filter_coefficient
            _5      complex
            _6      extrema
            _7      sequence_obsolete
            _8      centered_RIS
            _9      peak_detect
             endenum
;
<318>               PROCESSING_DONE: enum
             _0     no_processing
            _1      fir_filter
            _2      interpolated
            _3      sparsed
            _4      autoscaled
            _5      no_result
            _6      rolling
            _7      cumulative
             endenum
;
<320>          RESERVED5: word          ; expansion entry
;
<322>          RIS_SWEEPS: word         ; for RIS, the number of sweeps
                         ; else 1
;
; The following variables describe the basic acquisition
; conditions used when the waveform was acquired
;
<324>          TIMEBASE: enum
          _0  1_ps/div
          _1  2_ps/div
          _2  5_ps/div
          _3  10_ps/div
          _4  20_ps/div
          _5  50_ps/div
          _6  100_ps/div
          _7  200_ps/div
          _8  500_ps/div
```

_9 1_ns/div
_10 2_ns/div
_11 5_ns/div
_12 10_ns/div
_13 20_ns/div
_14 50_ns/div
_15 100_ns/div
_16 200_ns/div
_17 500_ns/div
_18 1_us/div
_19 2_us/div
_20 5_us/div
_21 10_us/div
_22 20_us/div
_23 50_us/div
_24 100_us/div
_25 200_us/div
_26 500_us/div
_27 1_ms/div
_28 2_ms/div
_29 5_ms/div
_30 10_ms/div
_31 20_ms/div
_32 50_ms/div
_33 100_ms/div
_34 200_ms/div
_35 500_ms/div
_36 1_s/div
_37 2_s/div
_38 5_s/div
_39 10_s/div
_40 20_s/div
_41 50_s/div
_42 100_s/div
_43 200_s/div
_44 500_s/div
_45 1_ks/div

```
          _46  2_ks/div
          _47  5_ks/div
              _100  EXTERNAL
            endenum
;
<326>              VERT_COUPLING: enum
            _0     DC_50_Ohms
           _1     ground
            _2     DC_1MOhm
           _3     ground
            _4     AC,_1MOhm
            endenum
;
<328>           PROBE_ATT: float
;
<332>              FIXED_VERT_GAIN: enum
            _0  1_uV/div
            _1  2_uV/div
            _2  5_uV/div
            _3  10_uV/div
            _4  20_uV/div
            _5  50_uV/div
            _6  100_uV/div
            _7  200_uV/div
            _8  500_uV/div
            _9  1_mV/div
            _10  2_mV/div
            _11  5_mV/div
            _12  10_mV/div
            _13  20_mV/div
            _14  50_mV/div
            _15  100_mV/div
            _16  200_mV/div
            _17  500_mV/div
            _18  1_V/div
            _19  2_V/div
            _20  5_V/div
```

```
            _21   10_V/div
            _22   20_V/div
            _23   50_V/div
            _24   100_V/div
            _25   200_V/div
            _26   500_V/div
            _27   1_kV/div
            endenum
;
<334>              BANDWIDTH_LIMIT: enum
          _0    off
          _1     on
            endenum
;
<336>             VERTICAL_VERNIER: float
;
<340>             ACQ_VERT_OFFSET: float
;
<344>              WAVE_SOURCE: enum
              _0     CHANNEL_1
              _1     CHANNEL_2
              _2     CHANNEL_3
              _3     CHANNEL_4
               _9     UNKNOWN
            endenum
;
/00                ENDBLOCK
;
;================================================================================================
;
USERTEXT: BLOCK
;
; Explanation of the descriptor block USERTEXT at most 160 bytes long.
;
;
<  0>    TEXT: text              ; a list of ASCII characters
;
```

```
/00              ENDBLOCK
;
;================================================================================
;
TRIGTIME: ARRAY
;
; Explanation of the trigger time array TRIGTIME.
; This optional time array is only present with SEQNCE waveforms.
; The following data block is repeated for each segment which makes up
; the acquired sequence record.
;
<  0>        TRIGGER_TIME: double     ; for sequence acquisitions,
                          ; time in seconds from first
                         ; trigger to this one
;
<  8>        TRIGGER_OFFSET: double   ; the trigger offset is in seconds
                          ; from trigger to zeroth data point
;
/00              ENDARRAY
;
;================================================================================
;
RISTIME: ARRAY
;
; Explanation of the random-interleaved-sampling (RIS) time array RISTIME.
; This optional time array is only present with RIS waveforms.
; This data block is repeated for each sweep which makes up the RIS record
;
<  0>       RIS_OFFSET: double       ; seconds from trigger to zeroth
                          ; point of segment
;
/00             ENDARRAY
;
;================================================================================
;
DATA_ARRAY_1: ARRAY
;
```

```
; Explanation of the data array DATA_ARRAY_1.
; This main data array is always present. It is the only data array for
; most waveforms.
; The data item is repeated for each acquired or computed data point
; of the first data array of any waveform.
;
<  0>        MEASUREMENT: data        ; the actual format of a data is
                              ; given in the WAVEDESC descriptor
                               ; by the COMM_TYPE variable.
;
/00               ENDARRAY
;
;================================================================================
;
DATA_ARRAY_2: ARRAY
;
; Explanation of the data array DATA_ARRAY_2.
; This is an optional secondary data array for special types of waveforms:
;       Complex FFT     imaginary part      (real part in DATA_ARRAY_1)
;      Extrema       floor trace        (roof trace in DATA_ARRAY_1)
;       Peak Detect    min/max pairs      (data values in DATA_ARRAY_1)
; In the first 2 cases, there is exactly one data item in DATA_ARRAY_2 for
; each data item in DATA_ARRAY_1.
; In Peak Detect waveforms, there may be fewer data values in DATA_ARRAY_2,
; as described by the variable POINTS_PER_PAIR.
;
<  0>       MEASUREMENT: data        ; the actual format of a data is
                             ; given in the WAVEDESC descriptor
                              ; by the COMM_TYPE variable.
;
/00              ENDARRAY
;
;================================================================================
;
SIMPLE: ARRAY
;
; Explanation of the data array SIMPLE.
```

```
; This data array is identical to DATA_ARRAY_1. SIMPLE is an accepted
; alias name for DATA_ARRAY_1.
;
<  0>      MEASUREMENT: data        ; the actual format of a data is
                          ; given in the WAVEDESC descriptor
                           ; by the COMM_TYPE variable.
;
/00            ENDARRAY
;
;================================================================================
;
DUAL: ARRAY
;
; Explanation of the DUAL array.
; This data array is identical to DATA_ARRAY_1, followed by DATA_ARRAY_2.
; DUAL is an accepted alias name for the combined arrays DATA_ARRAY_1 and
; DATA_ARRAY_2 (e.g. real and imaginary parts of an FFT).
;
<  0>         MEASUREMENT_1: data      ; data in DATA_ARRAY_1.
;
<  0>         MEASUREMENT_2: data      ; data in DATA_ARRAY_2.
;
/00            ENDARRAY
;
;
000000               ENDTEMPLATE
```

コマンドの使用

このセクションでは、オシロスコープが認識するコマンドとクエリをすべて説明します。説明は各コマンドごとに改ページされています。コマンド名（ヘッダ）は、ロングフォーマットとショートフォーマットの両方で記述されています。コマンド、クエリ、または両方のどれに該当するかも明記されています。クエリは、セットアップ・パラメータの変更や何らかの情報の取得など、アクションを促すコマンドです。クエリは、ヘッダ直後の疑問符（?）で識別します。

コマンドは、ロング・フォーマット名のアルファベット順に説明されます。したがって、ATTENUATIONのショート・フォーマットATTNとAUTO SETUPのショート・フォーマットASETを比較するとアルファベット順ではATTNの方が後ろですが、長いフォーマットが優先されるためATTENUATIONの説明が先です。それぞれのコマンド説明は、そのコマンドにより実行されるファンクションの簡単な説明から始まります。続いて、コマンドの正式な構文表記が、大文字と小文字で表したヘッダと、大文字のみで表した短いフォーマットで説明されます。クエリにも適用できる場合は、クエリの構文とその応答フォーマットが説明されます。

ほとんどの説明は、そのコマンドの一般使用の短い例示で終了します。GPIB例では、コントローラにナショナル・インスツルメンツのインタフェース・ボードが装着されているものと想定して、BASICの関連インタフェース・サブルーチンの呼び出しが示されます。オシロスコープのデバイス名は"SCOPE%"と定義されます。

☝ 記号は、特定なモデルやオプションでのみ使用できるコマンドで、その使用可能対象についての注があります。

以下のページの2つの特別索引表はマニュアル巻末の一般的な索引の補足ですが、これにより、このマニュアルが使いやすくなります。コマンド表Iでは、コマンド、クエリ、またはその両方がサブシステム別に分類されています。一方コマンド表IIでは、それぞれのコマンドがショート・フォーマット名のアルファベット順にリストされています。

コマンド表I　サブシステム別グループ分け

ﾍﾟｰｼﾞ	ショート	/ ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ	コマンドの機能説明
サブシステム ： 捕捉－波形捕捉を制御するコマンド			
15	ARM	**ARM_ACQUISITION**	信号捕捉待ち、捕捉ステータスを"停止(stopped)"より"シングル(single)"へ変更
19	ASET	**AUTO_SETUP**	入力信号に適した垂直、タイムベース、トリガの各パラメータの調節
16	ATTN	**ATTENUATION**	プローブの垂直減衰係数の選択
20	BWL	**BANDWIDTH_LIMIT**	帯域幅制限ローパス・フィルタのイネーブル/ディスエーブル
33	COMB	**COMBINE_CHANNELS**	チャネルインターリーブの設定
43	CPL	**COUPLING**	指定入力チャネルのカップリング・モード選択
83	GBWL	**GLOBAL_BWL**	全チャネルの帯域制限の設定
98	ILVD	**INTERLEAVED**	ランダム・インターリーブ・サンプリング(RIS: random interleaved sampling)のイネーブル/ディスエーブル
105	MSIZ	**MEMORY_SIZE**	最大メモリ長さ選択
108	OFST	**OFFSET**	入力チャネルの垂直オフセットの調節
138	PRDG	**PROBE_DEGAUSS**	電流プローブを消磁する
141	PRNA	**PROBE_NAME**	プローブの名前を返す
147	SCLK	**SAMPLE_CLOCK**	外部タイムベース制御を有効にする
152	SEQ	**SEQUENCE**	シーケンス・モード捕捉の条件設定
157	STOP	**STOP**	信号捕捉の即時停止
165	TDIV	**TIME_DIV**	タイムベース設定の変更
171	TRCP	**TRIG_COUPLING**	指定トリガ・ソースのカップリング・モード設定
172	TRDL	**TRIG_DELAY**	トリガポジション設定
170	*TRG	***TRG**	ARMコマンド実行
173	TRLV	**TRIG_LEVEL**	指定トリガ・ソースのトリガ・レベル調節
174	TRMD	**TRIG_MODE**	トリガ・モード指定
175	TRPA	**TRIG_PATTERN**	トリガ・パターンの定義
177	TRSE	**TRIG_SELECT**	捕捉トリガ条件選択
180	TRSL	**TRIG_SLOPE**	指定トリガ・ソースのトリガ・スロープ設定
181	TRWI	**TRIG_WINDOW**	カレント・エッジ・トリガ・ソースに窓振幅設定
186	VDIV	**VOLT_DIV**	垂直感度設定
188	WAIT	**WAIT**	現在の信号捕捉完了まで新規コマンド解析待機
サブシステム ： 通信－通信特性を設定するコマンド			
34	CFMT	**COMM_FORMAT**	波形データ送信フォーマットの選択
36	CHDR	**COMM_HEADER**	クエリ応答フォーマットの制御
38	CHL	**COMM_HELP_LOG**	リモート・コントロール・アシスタントが出力するログを返す

ﾍﾟｰｼﾞ	ｼｮｰﾄ	/ ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ	コマンドの機能説明
37	CHLP	**COMM_HELP**	ヘルプ診断のイネーブル
39	CONET	**COMM_NET**	オシロスコープのIPネットワーク・アドレスを指定
40	CORD	**COMM_ORDER**	波形データ転送のバイト・オーダの制御
41	CORS	**COMM_RS232**	RS-232-Cポートのリモート・コントロール・パラメータの設定

サブシステム ： カーソルー計測実行コマンド

44	CRMS	**CURSOR_MEASURE**	カーソル/パラメータ測定の表示タイプ指定
46	CRRD	**CURSOR_READOUT**	カーソルの読出し単位を設定
47	CRST	**CURSOR_SET**	8つのカーソルのうち任意の1カーソルの位置を設定する
49	CRVA	**CURSOR_VALUE**	指定カーソルで測定したトレース値を返す
102	MASK	**MASK**	Poly Maskの設定
109	OFCT	**OFFSET_CONSTANT**	オフセットをvoltsまたはdivisionに固定
113	PACL	**PARAMETER_CLR**	カスタム(Custom)、合否(Pass/Fail)のカレント・パラメータをすべてクリア
114	PACU	**PARAMETER_CUSTOM**	カスタムパラメータの設定
117	PADL	**PARAMETER_DELETE**	カスタム(Custom)、合否(Pass/Fail)の指定パラメータ削除
118	PAST	**PARAMETER_STATISTICS**	カレント統計パラメータ値を返す
119	PAVA	**PARAMETER_VALUE**	カレント・パラメータ、マスク試験値を返す
128	PECS	**PER_CURSOR_SET**	重ね書きカーソルの位置決定
130	PECV	**PER_CURSOR_VALUE**	重ね書きカーソルにより測定した値を返す
126	PFCO	**PASS_FAIL_CONDITION**	合否試験条件またはカスタム・パラメータの追加
122	PFCT	**PASS_FAIL_COUNTER**	合否捕捉カウンタのリセット
123	PFDO	**PASS_FAIL_DO**	返される合否試験結果内容と合否試験後のアクションの定義
124	PFMS	**PASS_FAIL_MASK**	トレースの許容範囲マスク生成と保存
125	PFST	**PASS_FAIL_STATUS**	指定行番号に対する合否試験ステータスを返す
195	XYCO	**XY_CURSOR_ORIGIN**	XYカーソル測定の原点を設定
196	XYCS	**XY_CURSOR_SET**	XY電圧カーソルの位置を設定する
198	XYCV	**XY_CURSOR_VALUE**	X対Yカーソルのカレント値を返す

サブシステム ： 表示ー波形を表示するコマンド

18	ASCR	**AUTO_SCROLL**	Auto Scroll機能を制御
24	CHST	**CALL_HOST**	サービス要求(SRQ: Service Request)の手動生成を有効にする
30	COLR	**COLOR**	画面上のオブジェクトそれぞれのカラー選択
32	CSCH	**COLOR_SCHEME**	表示カラー・スキーマの選択
63	DISP	**DISPLAY**	表示画面の制御
51	DPNT	**DATA_POINTS**	サンプル・ポイントの太字/シングル・ピクセル表示の制御
64	DTJN	**DOT_JOIN**	データ・ポイント間の補間線の制御
65	DZOM	**DUAL_ZOOM**	水平拡大率と位置の設定
72	FATC	**FAT_CURSOR**	カーソルの太さの指定
81	FSCR	**FULL_SCREEN**	グリッド用拡大表示フォーマットの選択

ﾍﾟｰｼﾞ	ショート	/ ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ	コマンドの機能説明
84	GRID	**GRID**	シングル・モード、デュアル・モード、またはクワッド・モード、オクタル・モードのグリッド表示指定
88	HMAG	**HOR_MAGNIFY**	選択した拡張トレースを水平方向に拡大
89	HPOS	**HOR_POSITION**	強調ゾーンの中心の水平方向位置決め
97	INTS	**INTENSITY**	グリッドまたはトレース/テキスト強度レベルの設定
100	KEY	**KEY**	メニュー・フィールドに文字列を表示
101	LOGO	**LOGO**	ロゴの表示/非表示の選択
104	MGAT	**MEASURE_GATE**	測定ゲート領域のハイライト制御
106	MSG	**MESSAGE**	メッセージ・フィールドに文字列表示
107	MZOM	**MULTI_ZOOM**	水平方向の拡大率および位置設定
132	PECL	**PERSIST_COLOR**	パーシスタンス・トレースのカラー・レンダリング方式の制御
133	PELT	**PERSIST_LAST**	パーシスタンス・データ・マップの最終トレースを表示
131	PERS	**PERSIST**	パーシスタンス表示モードのイネーブルまたはディスエーブル
134	PESA	**PERSIST_SAT**	パーシスタンスのカラー飽和レベル設定
135	PESU	**PERSIST_SETUP**	表示パーシスタンス期間の選択
150	SCSV	**SCREEN_SAVE**	自動スクリーン・セーバの制御
151	SEL	**SELECT**	手動表示制御の指定トレース選択
167	TOPA	**TRACE_OPACITY**	トレース・カラー透過性の制御
168	TORD	**TRACE_ORDER**	トレースの表示順を指定
166	TRA	**TRACE**	トレース表示のイネーブルまたはディスエーブル
184	VMAG	**VERT_MAGNIFY**	指定トレースを垂直方向に拡大
185	VPOS	**VERT_POSITION**	指定トレースの垂直方向位置を調節
194	XYAS	**XY_ASSIGN**	現在XYディスプレイに割り当てられているトレースを返す
200	XYDS	**XY_DISPLAY**	XYディスプレイ・モードのイネーブルまたはディスエーブル
201	XYRD	**XY_RENDER**	XY表示のシャープ/スムーズモードの選択
202	XYSA	**XY_SATURATION**	XY表示のパーシスタンスのカラー飽和レベル設定
サブシステム ： 関数－波形を数理処理するコマンド			
25	CLM	**CLEAR_MEMORY**	指定メモリのクリア
26	CLSW	**CLEAR_SWEEPS**	累積処理関数の再開始
55	DEF	**DEFINE**	関数評価用の数式指定
74	FCR	**FIND_CTR_RANGE**	累積イベントの最適表示
82	FRST	**FUNCTION_RESET**	波形処理関数のリセット
サブシステム ： ハード・コピー－表示画面の内容を印刷するするコマンド			
85	HCSU	**HARDCOPY_SETUP**	ハード・コピー・ドライバの構成
87	HCTR	**HARDCOPY_TRANSMIT**	ASCIIキャラクタの文字列をハード・コピー・ユニットへ送信
149	SCDP	**SCREEN_DUMP**	ハード・コピー・デバイスにスクリーン・ダンプ

ﾍﾟｰｼﾞ	ｼｮｰﾄ	ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ	コマンドの機能説明
サブシステム ： 保存装置－ディレクトリを作成し削除するコマンド			
60	DELF	**DELETE_FILE**	保存装置からファイルを削除
61	DIR	**DIRECTORY**	ファイル・ディレクトリの作成と削除
75	FCRD	**FORMAT_CARD**	メモリ・カードのフォーマッティング
77	FFLP	**FORMAT_FLOPPY**	フロッピー・ディスクのフォーマッティング
79	FHDD	**FORMAT_HDD**	取り外し可能ハード・ディスクのフォーマッティング
73	FLNM	**FILENAME**	デフォルト・ファイル名の変更
サブシステム ： その他－校正及びテストコマンド			
17	ACAL	**AUTO_CALIBRATE**	自動校正のイネーブルまたはディスエーブル
21	BUZZ	**BUZZER**	内蔵ブザーの制御
22	*CAL	***CAL**	内部校正の完全実施
23	COUT	**CAL_OUTPUT**	CAL BNCコネクタへの出力信号タイプ設定
52	DATE	**DATE**	内部リアル・タイム・クロックの日時変更
91	*IDN	***IDN**	機器情報の問い合わせ
111	*OPT	***OPT**	オシロスコープ・オプションの問い合わせ
137	PRCA	**PROBE_CAL**	電流プローブAP011の校正
139	PRIT	**PROBE_INFOTEXT**	プローブに関する情報テキストを返します
140	PRBL	**PROBE_LIGHTS**	アクティブプローブHFP1000のAutoColorID機能を設定
143	ROUT	**REAR_OUTPUT**	背面パネルの信号出力の設定
153	SLEEP	**SLEEP**	オシロスコープを指定時間スリープ
163	TDISP	**TDISP**	日付、時刻表示の設定
182	*TST	***TST**	内部自己試験実施
サブシステム ： セットアップの保存とリコール－・フロント・パネル設定情報を保存しリストアするコマンド			
112	PNSU	**PANEL_SETUP**	*SAV/*RSTコマンド補完、パネル設定の転送
142	*RCL	***RCL**	不揮発性パネル・セットアップ5種より1つをリコール
145	RCPN	**RECALL_PANEL**	保存装置からフロント・パネル・セットアップをリコール
146	*RST	***RST**	*RSTコマンドによるデバイス・リセット開始
148	*SAV	***SAV**	カレント状態を不揮発性内部メモリへ保存
159	STPN	**STORE_PANEL**	保存装置へのフロント・パネル・セットアップ情報の保存
サブシステム ： ステータス - ステータス情報を取り込みサービス要求をセットアップするコマンド			
14	ALST	**ALL_STATUS**	全ステータス・レジスタの読み出しとクリア
27	*CLS	***CLS**	全ステータス・データ・レジスタのクリア
28	CMR	**CMR**	コマンド・エラー・レジスタ(CMR: CoMmand error Register)の読み出しとクリア
53	DDR	**DDR**	デバイス依存レジスタ(DDR: Device Dependent Register)の読み出しとクリア

ﾍﾟｰｼﾞ	ショート	/ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ	コマンドの機能説明
66	EKEY	**ENABLE_KEY**	ローカルモードでのキーの使用
67	*ESE	***ESE**	標準イベント・ステータス・イネーブル(ESE: Event Status Enable)レジスタの設定
68	*ESR	***ESR**	イベント・ステータス・レジスタ(ESR: Event Status Register)の読み出しとクリア
70	EXR	**EXR**	実行エラー・レジスタ(EXR: EXecution error Register)の読み出しとクリア
92	INE	**INE**	内部ステート変更イネーブル(INE: Internal state change Enable)レジスタの設定
93	INR	**INR**	内部ステート変更レジスタ(INR: INternal state change Register)の読み出しとクリア
99	IST	**IST**	IEEE 488のカレント・ステートの読み出し
110	*OPC	***OPC**	イベント・ステータス・レジスタ(ESR: Event Status Register)にOPCビット設定
136	*PRE	***PRE**	パラレルポーリング・イネーブル(PRE: Parallel poll Enable)レジスタの設定
154	*SRE	***SRE**	サービスリクエストイネーブル(SRE: Service Request Enable)レジスタの設定
155	*STB	***STB**	IEEE 488内容の読み出し
183	URR	**URR**	ユーザリクエストステータス・レジスタ(URR: User Request status Register)の読み出しとクリア
187	*WAI	***WAI**	IEEE 488に必須

サブシステム ： 波形転送－波形を保存しリストアするコマンド

95	INSP	**INSPECT**	捕捉波形の部分読み出しを行なう
144	REC	**RECALL**	保存装置より内部メモリへ波形ファイルをリコール
158	STO	**STORE**	内部メモリまたは保存装置へのトレース保存
160	STST	**STORE_SETUP**	トレース保存方法の制御
162	STTM	**STORE_TEMPLATE**	保存装置への波形テンプレート保存
164	TMPL	**TEMPLATE**	完全な波形テンプレート・コピーの生成
169	TRFL	**TRANSFER_FILE**	オシロスコープとコンピュータ間のファイル転送
189	WF	**WAVEFORM**	コントローラよりオシロスコープへ波形転送
191	WFSU	**WAVEFORM_SETUP**	コントローラへの波形データ転送量を指定
193	WFTX	**WAVEFORM_TEXT**	波形のテキストドキュメント追加

コマンド表II　ショート・フォーマット名のアルファベット順

ﾍﾟｰｼﾞ	ｼｮｰﾄ	/ ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ		コマンドの機能説明
17	**ACAL**	AUTO_CALIBRATE	その他	自動校正のイネーブルまたはディスエーブル
14	**ALST**	ALL_STATUS	ｽﾃｰﾀｽ	全ステータス・レジスタの読み出しとクリア
15	**ARM**	ARM_ACQUISITION	捕捉	捕捉ステータスを"停止(stopped)"より"シングル(single)"へ変更
18	**ASCR**	AUTO_SCROLL	表示	Auto Scroll機能を制御
19	**ASET**	AUTO_SETUP	捕捉	垂直、タイムベース、トリガの各パラメータの調節
16	**ATTN**	ATTENUATION	捕捉	プローブの垂直減衰係数の選択
21	**BUZZ**	BUZZER	その他	内蔵ブザーの制御
20	**BWL**	BANDWIDTH_LIMIT	捕捉	帯域幅制御ローパス・フィルタのイネーブル/ディスエーブル
22	***CAL**	*CAL	その他	内部校正の完全実施
34	**CFMT**	COMM_FORMAT	通信	波形データ送信フォーマットの選択
36	**CHDR**	COMM_HEADER	通信	クエリ応答フォーマットの制御
38	**CHL**	COMM_HELP_LOG	通信	リモート・コントロール・アシスタントが出力するログを返す
37	**CHLP**	COMM_HELP	通信	ヘルプ診断のイネーブル
24	**CHST**	CALL_HOST	表示	サービスリクエスト(SRQ: Service Request)の手動生成を有効にする
25	**CLM**	CLEAR_MEMORY	関数	指定メモリのクリア
27	***CLS**	*CLS	ｽﾃｰﾀｽ	全ステータス・データ・レジスタのクリア
26	**CLSW**	CLEAR_SWEEPS	関数	累積処理関数の再開始
28	**CMR**	CMR	ｽﾃｰﾀｽ	コマンド・エラー・レジスタ(CMR: CoMmand error Register)の読み出しとクリア
30	**COLR**	COLOR	表示	画面上のオブジェクトそれぞれのカラー選択
33	**COMB**	COMBINE_CHANNELS	捕捉	チャネルインターリーブの設定
39	**CONET**	COMM_NET	通信	IPネットワーク・アドレスの指定
40	**CORD**	COMM_ORDER	通信	波形データ転送のバイト・オーダの制御
41	**CORS**	COMM_RS232	通信	RS-232-Cポートのリモート・コントロール・パラメータの設定
22	**COUT**	CAL_OUTPUT	その他	CAL BNCコネクタへの出力信号タイプ設定
43	**CPL**	COUPLING	捕捉	指定入力チャネルのカップリング・モード選択
44	**CRMS**	CURSOR_MEASURE	カーソル	カーソル/パラメータ測定のタイプ指定
46	**CRRD**	CURSOR_READOUT	カーソル	タイム・カーソルの振幅読出し単位(VまたはdBm)の設定
47	**CRST**	CURSOR_SET	カーソル	8つのカーソルのうち任意の1カーソルの位置決定を有効にする
49	**CRVA**	CURSOR_VALUE	カーソル	指定カーソルで測定したトレース値を返す

ﾍﾟｰｼﾞ	ｼｮｰﾄ / ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ			コマンドの機能説明
32	**CSCH**	COLOR_SCHEME	表示	表示カラー・スキーマの選択
52	**DATE**	DATE	その他	内部リアル・タイム・クロックの日時変更
53	**DDR**	DDR	ｽﾃｰﾀｽ	デバイス依存レジスタ(DDR: Device Dependent Register)の読み出しとクリア
55	**DEF**	DEFINE	関数	関数評価用の数式指定
60	**DELF**	DELETE_FILE	保存装置	保存装置からファイルを削除
61	**DIR**	DIRECTORY	保存装置	ファイル・ディレクトリの作成と削除
63	**DISP**	DISPLAY	表示	表示画面の制御
51	**DPNT**	DATA_POINTS	表示	サンプル・ポイントの太字/シングル・ピクセル表示の制御
64	**DTJN**	DOT_JOIN	表示	データ・ポイント間の補間線の制御
65	**DZOM**	DUAL_ZOOM	表示	水平拡大率と位置の設定
66	**EKEY**	ENABLE_KEY	ｽﾃｰﾀｽ	ローカルモードでのキーの使用
67	***ESE**	*ESE	ｽﾃｰﾀｽ	標準イベント・ステータス・イネーブル(ESE: Event Status Enable)レジスタの設定
68	***ESR**	*ESR	ｽﾃｰﾀｽ	イベント・ステータス・レジスタ(ESR: Event Status Register)の読み出しとクリア
70	**EXR**	EXR	ｽﾃｰﾀｽ	実行エラー・レジスタ(EXR: EXecution error Register)の読み出しとクリア
72	**FATC**	FAT_CURSOR	表示	カーソルの太さの指定
74	**FCR**	FIND_CTR_RANGE	関数	累積イベントの最適表示
75	**FCRD**	FORMAT_CARD	保存装置	メモリ・カードのフォーマッティング
77	**FFLP**	FORMAT_FLOPPY	保存装置	フロッピー・ディスクのフォーマッティング
79	**FHDD**	FORMAT_HDD	保存装置	取り外し可能ハード・ディスクのフォーマッティング
73	**FLNM**	FILENAME	保存装置	デフォルト・ファイル名の変更
82	**FRST**	FUNCTION_RESET	関数	波形処理関数のリセット
81	**FSCR**	FULL_SCREEN	表示	グリッド用拡大表示フォーマットの選択
83	**GBWL**	GLOBAL_BWL	捕捉	全チャネルの帯域制限の設定
84	**GRID**	GRID	表示	シングル・モード、デュアル・モード、またはクワッド・モードのグリッド表示指定
85	**HCSU**	HARDCOPY_SETUP	ﾊｰﾄﾞ・ｺﾋﾟｰ	ハード・コピー・ドライバの構成
87	**HCTR**	HARDCOPY_TRANSMIT	ﾊｰﾄﾞ・ｺﾋﾟｰ	ASCIIキャラクタの文字列をハード・コピー・ユニットへ送信
88	**HMAG**	HOR_MAGNIFY	表示	選択した拡張トレースを水平方向に拡大
89	**HPOS**	HOR_POSITION	表示	強調ゾーンの中心の水平方向位置決め
91	***IDN**	*IDN	その他	機器情報の問い合わせ
98	**ILVD**	INTERLEAVED	捕捉	ランダム・インターリーブ・サンプリング(RIS: random interleaved sampling)のイネーブル/ディスエーブル

ﾍﾟｰｼﾞ	ｼｮｰﾄ	/ ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ		コマンドの機能説明
92	**INE**	INE	ｽﾃｰﾀｽ	内部ステート変更イネーブル(INE: Internal state change Enable)レジスタの設定
93	**INR**	INR	ｽﾃｰﾀｽ	内部ステート変更レジスタ(INR: INternal state change Register)の読み出しとクリア
95	**INSP**	INSPECT	波形転送	捕捉波形の部分読み出しを有効にする
97	**INTS**	INTENSITY	表示	グリッドまたはトレース/テキスト強度レベルの設定
99	**IST**	IST	ｽﾃｰﾀｽ	IEEE 488のカレント・ステートの読み出し
100	**KEY**	KEY	表示	メニュー・フィールドに文字列を表示
101	**LOGO**	LOGO	表示	ロゴの表示/非表示の選択
102	**MASK**	MASK	カーソル	Poly Maskの設定
104	**MGAT**	MEASURE_GATE	表示	測定ゲート領域のハイライト制御
106	**MSG**	MESSAGE	表示	メッセージ・フィールドに文字列表示
105	**MSIZ**	MEMORY_SIZE	捕捉	最大メモリ長さ選択(MモデルとLモデルのみ)
107	**MZOM**	MULTI_ZOOM	表示	水平方向の拡大率および位置設定
109	**OFCT**	OFFSET_CONSTANT	カーソル	ゲイン変更時、オフセットvoltsまたはdivisionに固定
108	**OFST**	OFFSET	捕捉	出力チャネルの垂直オフセット調節を有効にする
110	***OPC**	*OPC	ｽﾃｰﾀｽ	イベント・ステータス・レジスタ(ESR: Event Status Register)にOPCビット設定
111	***OPT**	*OPT	その他	オシロスコープ・オプションの特定
113	**PACL**	PARAMETER_CLR	カーソル	カスタム(Custom)、合否(Pass/Fail)のカレント・パラメータをすべてクリア
114	**PACU**	PARAMETER_CUSTOM	カーソル	カスタマイズ可能クオリフィアのパラメータ制御
117	**PADL**	PARAMETER_DELETE	カーソル	カスタム(Custom)、合否(Pass/Fail)の指定パラメータ削除
118	**PAST**	PARAMETER_STATISTICS	カーソル	カレント統計パラメータ値を返す
119	**PAVA**	PARAMETER_VALUE	カーソル	カレント・パラメータ、マスク試験値を返す
132	**PECL**	PERSIST_COLOR	表示	パーシスタンス・トレースのカラー・レンダリング方式の制御
128	**PECS**	PER_CURSOR_SET	カーソル	独立カーソルの位置決定
130	**PECV**	PER_CURSOR_VALUE	カーソル	カーソルにより測定した値を返す
133	**PELT**	PERSIST_LAST	表示	パーシスタンス・データ・マップの最終トレースを表示
131	**PERS**	PERSIST	表示	パーシスタンス表示モードのイネーブルまたはディスエーブル
134	**PESA**	PERSIST_SAT	表示	パーシスタンスのカラー飽和レベル設定
135	**PESU**	PERSIST_SETUP	表示	表示パーシスタンス期間の選択
126	**PFCO**	PASS_FAIL_CONDITION	カーソル	合否試験条件またはカスタム・パラメータの追加
122	**PFCT**	PASS_FAIL_COUNTER	カーソル	合否捕捉カウンタのリセット
123	**PFDO**	PASS_FAIL_DO	カーソル	返される合否試験結果内容と合否試験後のアクションの定義

ﾍﾟｰｼﾞ	ｼｮｰﾄ	/ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ		コマンドの機能説明
124	**PFMS**	PASS_FAIL_MASK	カーソル	トレースの許容範囲マスク生成と保存
125	**PFST**	PASS_FAIL_STATUS	カーソル	指定行番号に対する合否試験ステータスを返す
112	**PNSU**	PANEL_SETUP	保存/ﾘｺｰﾙ	*SAV/*RSTコマンド補完
140	**PRBL**	PROBE_LIGHTS	プローブ	アクティブプローブHFP1000のAutoColorID機能を設定
137	**PRCA**	PROBE_CAL	その他	電流プローブAP011の校正
138	**PRDG**	PROBE_DEGAUSS	捕捉	電流プローブを消磁する
136	***PRE**	*PRE	ｽﾃｰﾀｽ	パラレルポーリング・イネーブル(PRE: Parallel poll Enable)レジスタの設定
139	**PRIT**	PROBE_INFOTEXT	プローブ	Waverunnerに接続されたプローブに関する情報を返す
141	**PRNA**	PROBE_NAME	捕捉	プローブの名前を返す
142	***RCL**	*RCL	保存/ﾘｺｰﾙ	不揮発性パネル・セットアップ5種より1つをリコール
145	**RCPN**	RECALL_PANEL	保存/ﾘｺｰﾙ	保存装置からフロント・パネル・セットアップをリコール
144	**REC**	RECALL	波形転送	保存装置より内部メモリへファイルをリコール
143	**ROUT**	REAR_OUTPUT	その他	背面パネルの信号出力の設定
146	***RST**	*RST	保存/ﾘｺｰﾙ	*RSTコマンドによるデバイス・リセット開始
148	***SAV**	*SAV	保存/ﾘｺｰﾙ	カレント状態を不揮発性内部メモリへ保存
149	**SCDP**	SCREEN_DUMP	ﾊｰﾄﾞ・ｺﾋﾟｰ	ハード・コピー・デバイスにスクリーン・ダンプ
147	**SCLK**	SAMPLE_CLOCK	捕捉	外部タイムベース制御を有効にする
150	**SCSV**	SCREEN_SAVE	表示	自動スクリーン・セーバの制御
151	**SEL**	SELECT	表示	手動表示制御の指定トレース選択
152	**SEQ**	SEQUENCE	捕捉	シーケンス・モード捕捉の条件設定
153	**SLEEP**	SLEEP	その他	オシロスコープを指定時間スリープ
154	***SRE**	*SRE	ｽﾃｰﾀｽ	サービス・リクエスト・イネーブル(SRE: Service Request Enable)レジスタの設定
155	***STB**	*STB	ｽﾃｰﾀｽ	IEEE 488内容の読み出し
158	**STO**	STORE	波形転送	内部メモリまたは保存装置へのトレース保存
157	**STOP**	STOP	捕捉	信号捕捉の即時停止
159	**STPN**	STORE_PANEL	保存/ﾘｺｰﾙ	保存装置へのフロント・パネル・セットアップ情報の保存
160	**STST**	STORE_SETUP	波形転送	トレース保存方法の制御
162	**STTM**	STORE_TEMPLATE	波形転送	保存装置への波形テンプレート保存
163	**TDISP**	TDISP	その他	日付、時刻表示の設定
165	**TDIV**	TIME_DIV	捕捉	タイムベース設定の変更
164	**TMPL**	TEMPLATE	波形転送	完全な波形テンプレート・コピーの生成
167	**TOPA**	TRACE_OPACITY	表示	トレース・カラー透過性の制御
168	**TORD**	TRACE_ORDER	表示	トレースの表示順
166	**TRA**	TRACE	表示	トレース表示のイネーブルまたはディスエーブル
171	**TRCP**	TRIG_COUPLING	捕捉	指定トリガ・ソースのカップリング・モード設定

ﾍﾟｰｼﾞ	ｼｮｰﾄ	/ﾛﾝｸﾞ・ﾌｫｰﾏｯﾄ		コマンドの機能説明
172	**TRDL**	TRIG_DELAY	捕捉	トリガ発生時刻設定
169	**TRFL**	TRANSFER_FILE	波形ｽﾄﾚｰｼﾞ	ストレージ・メディアとのファイル転送、オシロスコープとコンピュータ間のファイルの転送
170	***TRG**	*TRG	捕捉	ARMコマンド実行
173	**TRLV**	TRIG_LEVEL	捕捉	指定トリガ・ソースのトリガ・レベル調節
174	**TRMD**	TRIG_MODE	捕捉	トリガ・モード指定
175	**TRPA**	TRIG_PATTERN	捕捉	トリガ・パターンを定義
177	**TRSE**	TRIG_SELECT	捕捉	捕捉トリガ条件選択
180	**TRSL**	TRIG_SLOPE	捕捉	指定トリガ・ソースのトリガ・スロープ設定
181	**TRWI**	TRIG_WINDOW	捕捉	カレント・エッジ・トリガ・ソースに窓振幅設定
182	***TST**	*TST	その他	内部自己試験実施
183	**URR**	URR	ｽﾃｰﾀｽ	ユーザ・リクエスト・ステータス・レジスタ(URR: User Request status Register)の読み出しとクリア
186	**VDIV**	VOLT_DIV	捕捉	垂直感度設定
184	**VMAG**	VERT_MAGNIFY	表示	指定トレースを垂直方向に拡大
185	**VPOS**	VERT_POSITION	表示	指定トレースの垂直方向位置を調節
187	***WAI**	*WAI	ｽﾃｰﾀｽ	IEEE 488に必須
188	**WAIT**	WAIT	捕捉	カレント解析完了まで新規解析待機
189	**WF**	WAVEFORM	波形転送	コントローラよりオシロスコープへ波形転送
191	**WFSU**	WAVEFORM_SETUP	波形転送	コントローラへの波形データ転送量を指定
193	**WFTX**	WAVEFORM_TEXT	波形転送	捕捉条件のドキュメント化
194	**XYAS**	XY_ASSIGN	表示	現在XYディスプレイに割り当てられているトレースを返す
195	**XYCO**	XY_CURSOR_ORIGIN	カーソル	絶対カーソル測定の原点を設定
196	**XYCS**	XY_CURSOR_SET	カーソル	XY電圧カーソルの位置決定を有効にする
198	**XYCV**	XY_CURSOR_VALUE	カーソル	X対Yカーソルのカレント値を返す
200	**XYDS**	XY_DISPLAY	表示	XYディスプレイ・モードのイネーブルまたはディスエーブル
201	**XYRD**	XY_RENDER	表示	XY表示のシャープ/スムーズモードの選択
202	**XYSA**	XY_SATURATION	表示	XY表示のパーシスタンスのカラー飽和レベルの設定

コマンド実行

オシロスコープでは、コマンドまたはクエリの実行前に、そのコマンドまたはクエリが正しいか、またその実行に十分な情報が与えられているか検証されます。ローカル・ユーザが制御できないオシロスコープの挙動を、ローカル・ユーザによって変更されることのないよう、リモート・ユーザはオシロスコープをリモート・ステートに設定して、オシロスコープを制御するコマンドを実行する必要があります。オシロスコープがローカル・ステートで動作しているときにこのようなコマンドを送信すると、実行許可エラーが発生し、そのコマンド実行は拒否されます。同様に、オシロスコープがリモート・ステートで動作しているときはフロント・パネル制御はすべてディスエーブルされており、ローカル・ユーザはリモート・ユーザに割り込むことはできません。

オシロスコープに問い合わせを行ってもその内部ステートは変化しないので、オシロスコープがローカル・ステートであるかリモート・ステートであるかにかかわらず、常に問い合わせできます。例外は、*CAL?クエリと*TST?クエリの2つのみです。これらはともにオシロスコープを再校正するので、リモート・ステートでのみ実行されます。

リモート挙動にのみ影響を与えるコマンドは、オシロスコープがローカル・ステートであるかリモート・ステートであるかにかかわらず実行されます。これらには、通信パラメータを変更するコマンドすべて（COMM_FORMAT、COMM_HEADER、COMM_HELP、COMM_ORDER、COMM_RS232）、ステータス情報に影響を与えるコマンドすべて（*CLS、*ESE、INE、*OPC、*PRE、*SRE、*WAI）、ローカルユーザへの画面メッセージ表示に使用されるコマンドす（CALL_HOST、KEY、MESSAGE）が含まれます。

ここで説明したものは、コマンドはリモート・ステートでのみ実行され、クエリはローカル・ステートとリモート・ステートの両方で実行される、という原則にあてはまらない例外のみです。

コマンド表記

以降の説明には次の表記を用います。

<> プレース・ホールダーとして用いられる語は角括弧で括ります。プレース・ホールダーには、ヘッダ・パスと、コマンドのデータ・パラメータの2タイプがあります。

:= コロンの後に等号を付けて、プレース・ホールダと、プレース・ホールダの代わりにコマンドで使用できる値のタイプや範囲の記述とを分離します。

{} 必ず1つ選択する必要のある選択肢のリストを中括弧で囲みます。

[] オプション項目を大括弧で囲みます。

... 省略記号は、その左右両方に対して項目を複数回繰り返し可能であることを示します。

例として、垂直入力感度設定コマンドの構文表記を以下に示します。

<channel>:VOLT_DIV<v_gain>

<channel>:={C1, C2}

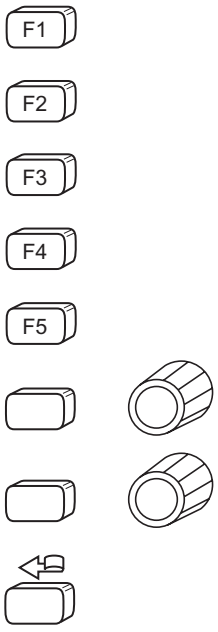
<v_gain>:=5.0 mV ～ 2.5V

最初の行は、コマンドの正式表現です。<channel>は、ヘッダ・パスのプレース・ホールダーを表し、<v_gain>は目的の垂直ゲイン値を指定するデータ・パラメータのプレース・ホールダーです。2行目は、"C1"または"C2"のどちらかをヘッダ・パスに選択する必要があることを示しています。3行目は、実際の垂直ゲインを5 mV～2.5 Vの範囲で任意に設定できることを示しています。

コマンド

以降およそページ200にわたって、140あまりのコマンド、クエリ、およびその両方に属すものを個別に説明します。

オシロスコープのリセット



オシロスコープのリセット：電源投入時のデフォルト状態に戻すには、下記ボタンを同時に押します。

ステータス　ALL_STATUS?、ALST?

クエリ

説　明

ALL_STATUS?クエリは、STBレジスタのMAVビット（ビット6）を除き、STB、ESR、INR、DDR、CMR、EXR、URRの全ステータス・レジスタの内容を読み出しクリアします。各レジスタの内容解釈については、該当するステータス・レジスタを参照してください。

ALL_STATUS?クエリは、オシロスコープのステート概要をすべて表示するのに便利です。

クエリ構文

ALI_STatus?

応答フォーマット

ALI_STatus STB, <value>, ESR, <value>, INR, <value>,
DDR, <value>, CMR, <value>, EXR, <value>, URR, <value>
<value>:= 0～65535

例（GPIB）

次の命令は、全ステータス・レジスタの内容を読み出します。

```
CMD$="ALST?":CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
```

応答メッセージ:

```
ALST STB, 000000, ESR, 000052, INR, 000005, DDR, 000000,
CMR, 000004, EXR, 000024, URR, 000000
```

関連コマンド

*CLS, CMR?, DDR?, *ESR?, EXR?, *STB?, URR?

捕　捉

ARM_ACQUISITION、ARM コマンド

説　明

ARM_ACQUISITIONコマンドは、捕捉ステートを"停止（stopped）"より"シングル（single）"へ変更して、信号捕捉プロセスをイネーブルにします。

コマンド構文

ARM_acquisition

例

次のコマンドは、信号捕捉をイネーブルにします。

```
CMD$="ARM":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

STOP, *TRG, TRIG_MODE, WAIT

捕　捉

ATTENUATION、ATTN

コマンド / クエリ

説　明

ATTENUATIONコマンドは、プローブの垂直減衰係数を選択します。指定できる値は、1、2、5、10、20、50、100、200、500、1000、10000です。

ATTENUATION?クエリは、指定チャネルの減衰係数を返します。

コマンド構文

<channel>:ATTeNuation <attenuation>

<channel>:={C1, C2, C3☝, C4☝, EX☝, EX10☝}

<attenuation>:={1, 2, 5, 10, 20, 25, 50, 100, 200, 500, 1000, 10000}

クエリ構文

<channel>:ATTeNuation?

応答フォーマット

<channel>:ATTeNuation <attenuation>

☝使用可能対象

<channel>:C3およびC4－4チャネル・オシロスコープでのみ使用できます。

<channel>:EX, EX10－クエリのみ使用できます。

例（GPIB）

次のコマンドは、チャネル1の減衰係数を100に設定します。

```
CMD$="C1:ATTN 100":CALL IBWRT (SCOPE%, CMD$)
```

AUTO_CALIBRATE、ACAL

コマンド / クエリ

説　明

AUTO_CALIBRATEコマンドは、オシロスコープの自動校正をイネーブルまたはディスエーブルします。電源投入時には、自動校正がON、つまりVdto/Divの設定とTime/Div設定に関して全入力チャネルが定期的に校正されます。

自動校正は、ACAL　OFFコマンドを発行してディスエーブルにできます。随時*CAL?クエリを発行してオシロスコープを完全に校正できます。オシロスコープをローカル・コントロールに戻すと、定期校正が再開されます。

AUTO_CALIBRATE?クエリに対する応答では、自動校正がイネーブルかどうかが示されます。

コマンド構文

Auto_CALibrate <state>

<state>:={ON, OFF}

クエリ構文

Auto_CALibrate?

応答フォーマット

Auto_CALibrate <state>

例（GPIB）

次の命令は、自動校正をディスエーブルにします。

```
CMD$="ACAL  OFF":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

*CAL?

表　示　　AUTO_SCROLL、ASCR

コマンド / クエリ

説　明

AUTO_SCROLLコマンドおよびクエリは、Auto Scroll機能を制御します。このコマンドには、フロント・パネルの「MATH SETUP」ボタンと「ZOOM+MATH」メニューを使用してアクセスします。

このコマンドは、自動的に、選択したトレース(または、マルチ・ズームがオンの場合、すべてのトレース)を画面からスクロールします。コマンドは、スクロールをオン/オフにし、スクロール速度をディビジョン数/秒で設定します。クエリは、現在のスクロール速度を返します。

コマンド構文

Auto_SCRoll <action>,<mode>,<speed>

<action>:={PLAY,REVERSE,STOP}

<mode>:={DIV_S,DIV_U}

注: DIV_S:div/s 0.1～50 DIV_U:number of div 0.01～10

<SPEED>:=DIV_Uモードの場合、0.01-10.00

DIV_Sモードの場合、0.10-10.00

クエリ構文

Auto_SCRoll?

応答フォーマット

Auto_SCRoll <action>,<mode>,<speed>

例(GPIB)

次のコマンドはAuto Scrollをオンにし、2s/divの速度でデータを右にスクロールし始めます。

```
CMD$="ASCR PLAY,DIV_S,2":CALL IBWRT(SCOPE%,CMD$)
```

関連コマンド

MULTI_ZOOM, HOR_MAGNIFY, HOR_POSITION

捕　捉　　AUTO_SETUP、ASET コマンド

説　明

AUTO_SETUPコマンドは、垂直、タイムベース、トリガの各パラメータを調節して入力信号を表示します。AUTO_SETUPは、現在トレースがオンされているチャネルでのみ動作します。どのトレースもオンされていない場合は、AUTO_SETUPはすべてのチャネルで動作し、それらのトレースすべてをオンにします。

複数のチャネルで信号が検出されると、そのチャネルのうち番号の一番小さいチャネルに対してタイムベースとトリガ・ソースが選択されます。

1入力チャネルのみオン状態の場合は、タイムベースはそのチャネルに対して調節されます。

<channel>:AUTO_SETUP FINDコマンドは、指定チャネルに対してのみゲインとオフセットを調節します。

コマンド構文

<channel>:Auto_SETup [FIND]

<channel>:{C1, C2, C3☝, C4☝}

FINDキーワードがあると、ゲインとオフセットの調節は指定チャネルでのみ実行され、FINDキーワードがない場合は、接頭文字列<channel>のあるなしにかかわらず通常のオート・セットアップが実行されます。

☝使用可能対象

<channel>:C3およびC4－4チャネル・オシロスコープでのみ使用できます。

例

次のコマンドは、オシロスコープによるオート・セットアップを実行します。

```
CMD$="ASET":CALL IBWRT (SCOPE%, CMD$)
```

捕　捉

BANDWIDTH_LIMIT、BWL☝

コマンド / クエリ

説　明

BANDWIDTH_LIMITコマンドは、帯域幅制限ローパス・フィルタをイネーブルまたはディスエーブルします。

BANDWIDTH_LIMIT?クエリの応答は、帯域幅フィルタのオン/オフを示します。

コマンド構文

<channel>:BandWidth_Limit <mode>

<mode>:={ON, OFF, 200MHZ}☝

クエリ構文

BandWidth_Limit?

応答フォーマット

BandWidth_Limit <mode>

☝使用可能対象

203ページのモデル別：使用可能対象一覧をご参照ください。

例

次のコマンドは帯域幅フィルタをオンにします。

```
CMD$="BWL ON":CALL IBWRT  (SCOPE%, CMD$)
```

BUZZER、BUZZ

コマンド

説　明

BUZZERコマンドは、内蔵ブザーを制御します。これは、対話型作業アプリケーションのローカル・オペレータの注意を惹くのに有用です。ブザーは、ユーザ選択するブザーのON/OFFにより、短い音（BEEPモードで約400 ms）をまたは一定の時間間隔で連続的に鳴らすことができます。このコマンドは、CLBZハード・オプション適用オシロスコープでのみ使用できます。

注: このコマンドは、（ローカルでもリモートでも）常に受け付けられます。

コマンド構文

BUZZer <state>

<state>:={BEEP, ON, OFF}

例（GPIB）

次のコードを送ると、オシロスコープで短い音が2回鳴ります。

```
CMD$="BUZZ BEEP;BUZZ BEEP":
CALL IBWRT (SCOPE%, CMD$)
```

その他

*CAL?

クエリ

説　明

*CAL?クエリは、内部校正を完全実行します。この校正シーケンスは、電源投入時の校正シーケンスと同じです。校正終了時には、どのように校正が終了したかが応答され、オシロスコープはクエリの前のステートに戻されます。

クエリ構文

*CAL?

応答フォーマット

*CAL <diagnostics>

<diagnostics>:=0 or 1

0 = 障害なし

☝使用可能対象

オシロスコープがリモート・モードの場合にのみ使用できます。

例（GPIB）

次の命令は、自己校正を実行します。

CMD$="*CAL?:CALL IBWRT（SCOPE%, CMD$）:

応答メッセージ（障害のない場合）: *CAL 0

関連コマンド

AUTO_CALIBRATE

説　明

CAL_OUTPUT コマンドは、CAL端子へ出力する信号の種類を設定します。

コマンド構文

Cal_OUTput <mode>[,<level>[,<rate>]]
<mode> : = {CALSQ, LEVEL}
<level> : = (1.0 to 1.00 V (1 M(に対して)
<rate> : = 500 Hz～1 MHz.

クエリ構文

Cal_OUTput?

応答フォーマット

Cal_OUTput <mode>,<level>[,<rate>]

☝使用可能対象

コマンドを送出する時、<rate>は、<mode>がCALSQの場合のみ受け付けます。なお、LC及び93xxシリーズでサポートされていた<mode>:=OFF,PF,TRIG,TRDYは、REAR_OUTPUTコマンドでサポートします。

例（GPIB）

以下のコマンドは、10KHで0.2Vの振幅を持つ方形波を校正信号として出力するように設定します。

```
CMD$="COUT CALSQ,0.2 V,10 kHz":
CALL IBWRT(SCOPE%,CMD$)
```

関連コマンド

REAR_OUTPUT

補足情報

表　　記	
CALSQ	方形波を出力します。
LEVEL	直流信号を出力します。

表　示　　CALL_HOST、CHST

コマンド / クエリ

説　明

CALL_HOSTコマンドにより、ユーザは手動でサービス・リクエスト（SRQ: Service Request）を生成できます。CALL_HOSTコマンドが受信されると、"Call Host（ホスト呼び出し）"メッセージが、画面のすぐ隣のメニュー・ボタン・カラムの一番下のボタンの隣に表示されます。ルート・メニューでこのボタンを押すと、ユーザ・リクエスト・ステータス・レジスタ（URR: User Request status Register）とイベント・ステータス・レジスタのURQビットが設定されます。これにより、サービス・リクエスト・メカニズムがイネーブルになっているとき、ローカル・モードでSRQが生成できます。

CALL_HOST?クエリに対する応答は、CALL_HOSTがイネーブル（ON）であるかディスエーブル（OFF）であるかを示します。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

Call_HoST <state>

<state>:={ON, OFF}

クエリ構文

Call _HoST?

応答フォーマット

Call_HoST?

例（GPIB）

次のコード実行後は、ボタンを押すたびにSRQリクエストが生成されます。SRQサービスがイネーブルになっていると想定しています。

```
CMD$="CHST ON":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

URR

説　明

CLEAR_MEMORYコマンドは、指定メモリをクリアします。このメモリにそれまでに保存されたデータは消去され、メモリ空間が空きメモリ・プールに返されます。

コマンド構文

CLear_Memory <memory>

<memory>:={M1, M2, M3, M4}

例（GPIB）

次のコマンドは、メモリM2をクリアします。

CMD$="CLM M2":CALL IBWRT（SCOPE%, CMD$）

関連コマンド

STORE

関　数　CLEAR_SWEEPS、CLSW

コマンド

説　明

CLEAR_SWEEPSコマンドは、加算平均または連続平均、Extrema（極値）、FFT乗数平均、ヒストグラム、パルス・パラメータ統計、合否カウンタ、パーシスタンスなどの累積処理関数のカウントをクリアしスタートします。

コマンド構文

CLear_SWeeps

例（GPIB）

次の例は、累積処理関数を再開始します。

```
CMD$="CLSW":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

DEFINE, INR

説　明

*CLSコマンドは、全ステータス・データ・レジスタをクリアします。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

*CLS

例（GPIB）

次のコマンドは、全ステータス・データ・レジスタをクリアします。

```
CMD$="*CLS":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

ALL_STATUS, CMR, DDR, *ESR, EXR, *STB, URR

ステータス

CMR?

クエリ

説　明

CMR?クエリは、オシロスコープが検出した最終構文エラー・タイプを示すコマンド・エラー・レジスタ（CMR: CoMmand error Register、次の表を参照してください）の内容を読み出し、クリアします。

クエリ構文

CMR?

応答フォーマット

CMR <value>

<value>:=0～13

例（GPIB）

次の命令は、CMRレジスタの内容を読み出します。

```
CMD$="CMR?":CALL IBWRT (SCOPE%, CMD$)
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
応答メッセージ:
CMR 0
```

関連コマンド

ALL_STATUS?, *CLS

補足情報

値	説　　明
1	コマンド / クエリ・ヘッダ認識不能
2	ヘッダ・パス不正
3	番号不正
4	接尾番号不正
5	キーワード認識不能
6	文字列エラー
7	別のメッセージに埋め込まれた GET
10	任意データ・ブロックが予定されている
11	任意データ・ブロックのバイト・カウント・フィールドに数字以外の文字がある
12	定義長データ・ブロック転送中 EOI 検出
13	定義長データ・ブロック転送中バイト溢れ検出

コマンド・エラー・ステータス・レジスタ（CMR: Command Error Status Register）構造

表　示

COLOR、COLR☝

コマンド / クエリ

説　明

COLORコマンドは、テキスト、トレース、グリッド、カーソルなど個々の表示オブジェクトのカラーを選択します。

COLOR?クエリに対する応答は、現在表示されているかいないかにかかわらず、各表示オブジェクトに割り当てられたカラーを示します。

コマンド構文

COLoR <object, color>[,...<object>, <color>]

<object>:={BACKGND, C1, C2, C3, C4, TA, TB, TC, TD, GRID, TEXT, CURSOR, NEUTRAL, WARNING,}

<color>:={WHITE, CYAN, YELLOW, GREEN, MAGENTA, BLUE, RED, LTGRAY, GRAY, SLGRAY, CHGRAY, DKCYAN, CREAM, SAND, AMBER, OLIVE, LTGEEN, JADE, LMGREEN, APGREEN, EMGREEN, GREREEN, OCSPRAY, ICEBLUE, PASTBLUE, PALEBLUE, SKYBLUE, ROYLBLUE, DEEPBLUE, NAVY, PLUM, PURPLE, AMETHYST, FUCHSIA, RASPBRY, NEONPINK, PALEPINK, PINK, VERMIL, ORANGE, CERISE, KHAKI, BROWN, BLACK}

クエリ構文

COLoR?

応答フォーマット

COLoR <object>, <color>[,...<object>, <color>]

☝使用可能対象

このコマンド／クリエは、COLOR_SCHEMEコマンドで、色設定が、ユーザー設定可能な状態 (U1～U4) になっている時のみ受けつけます。

例 (GPIB)

次のコマンドは、チャネル1のカラーに赤を選択します。

```
CMD$="COLR, RED":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

COLOR_SCHEME, PERSIST_COLOR

補足情報

表　記			
<color>	**カラー**	**<color>**	**カラー**
WHITE	白	OCSPRAY	オーシャン・スプレイ
CYAN	シアン	ICEBLUE	淡青色
YELLOW	黄	PASTBLUE	パステル・ブルー
GREEN	緑	PALEBLUE	薄青
MAGENTA	マゼンタ	SKYBLUE	青空
BLUE	青	ROYLBLUE	藤紫
RED	赤	DEEPBLUE	深青
LTGRAY	薄いグレー	NAVY	ネイビー
GRAY	グレー	PLUM	暗紫
SLGRAY	ｽﾚｰﾄ（黄緑がかったｸﾞﾚｰ）	PURPLE	紫
CHGRAY	黒灰（ﾁｬｺｰﾙ･ｸﾞﾚｰ）	AMETHYST	アメジスト（紫水晶）
DKCYAN	暗いシアン	FUCHSIA	明るい（赤がかった）紫
CREAM	クリーム	RASPB	濃い赤紫（ラズベリー）
SAND	砂色	NEONPINK	ネオン・ピンク
AMBER	琥珀	PALEPINK	薄桃
OLIVE	オリーブ	PINK	ピンク
LTGREEN	薄青	VEGRMIL	朱
JADE	翡翠	OGRANGE	橙
LMGGREEN	ライム	CEGRE	鮮紅
APGGREEN	青リンゴ	KHAKI	カーキ色
EMGGREEN	鮮緑	BROWN	茶
GRGREEN	黄緑	BLACK	黒

<object>	**表示オブジェクト**	**<object>**	**表示オブジェクト**
BACKGND	背景	CURSOR	カーソル
C1..C4	チャネル・トレース	WARNING	警告メッセージ
TA..TD	関数トレース	NEUTRAL	ニュートラル・カラー
GRID	グリッド・ライン	OVERLAYS	ﾒﾆｭｰ背景色（ﾌﾙｽｸﾘｰﾝ）

表　示

COLOR_SCHEME、CSCH

コマンド / クエリ

説　明

COLOR_SCHEMEコマンドは、表示カラー・スキーマを選択します。

COLOR_SCHEME?クエリに対する応答は、使用するカラー・スキーマを示します。

コマンド構文

Color_SCHeme <scheme>

<scheme>:={1, 2, 3, 4, 5, 6, 7, U1, U2, U3, U4}

クエリ構文

Color_SCHeme?

応答フォーマット

Color_SCHeme <scheme>

例（GPIB）

次の命令は、ユーザ・カラー・スキーマU2を選択します。

```
CMD$="CSCH U2":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

COLOR, PERSIST_COLOR

捕　捉

COMBINE_CHANNELS、COMB

コマンド / クエリ

説　明

COMBINE_CHANNELSコマンドは、チャネルインタリーブ機能のチャネルを設定します。

"2"を選択すると、CH1とCH2、CH3とCH4、のペアで結合し、CH2とCH3へ入力した信号が捕捉できます。"Auto"を選択すると表示チャネルが2チャネル以下の場合、CH1とCH2、CH3とCH4のペアで結合し、CH1またはCH2、CH3またはCH4へ入力した信号が捕捉できます。

COMBINE_CHANNELS?クエリは、チャネルインタリーブで現在使用しているチャネルを返します。

コマンド構文

COMBine_channels <state>

<state>: ={1,2,Auto}

クエリ構文

COMBine_channels?

応答フォーマット

COMB <state>

☝使用可能対象

203ページのモデル別：使用可能対象一覧をご参照ください。

例（GPIB）

次のコマンドは CH1とCH2、CH3とCH4間のチャネルインタリーブを設定します。

```
CMD$="COMB 2":CALL IBWRT (SCOPE%, CMD$)
```

通　信　　COMM_FORMAT、CFMT

コマンド / クエリ

説　明

COMM_FORMATコマンドは、オシロスコープが波形データ送信に使用するフォーマットを選択します。オプションで、ブロック・フォーマット、データ・タイプ、符号化モードのデフォルト設定を変更できます。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

COMM_FORMAT?クエリは、現在選択されている波形データ・フォーマットを返します。

コマンド構文

Comm_ForMaT <block_formt>, <data_type>, <encoding>

<block_format>:={DEF9, INDO, OFF}

<data_type>:={BYTE, WORD}

<encoding>:={BIN, HEX}

(GPIBでは両方の符号化フォーマットが使用され、RS-232-Cでは常にHEXが使用されます)。

初期設定（電源投入時）は以下のように設定されています。

DEF9, WORD, BIN for GPIB

DEF9, WORD, HEX for RS-232-C

クエリ構文

Comm_ForMaT?

応答フォーマット

Comm_ForMaT <block_format>, <data_type>, <encoding>

例（GPIB）

次のコードは、波形データの転送フォーマットを再定義します。データは、未定義長の1ブロックで転送されます。データは、バイナリに符号化され8ビット整数で表わされます。

CMD$="CFMT INDO, BYTE, BIN":CALL IBWRT (SCOPE%, CMD$)

補足情報

ブロック・フォーマット

DEF9: IEEE 488.2定義任意長ブロック応答データ・フォーマットを使用します。数字9はバイト・カウントが9桁で構成されることを示します。データ・ブロックは、バイト・カウント・フィールド直後に続きます。

たとえば、3データ・バイトで構成されるデータ・ブロックは、以下のように送信されます。

WF DAT1, #9000000003<DAB><DAB><DAB>

ここで、<DAB>は8ビットのバイナリ・データ・バイトを示します。

INDO: IEEE 488.2未定義任意長ブロック応答データ・フォーマットを使用します。

<NL^END>（EOIで改行）はブロック転送完了を示します。

上と同じデータ・バイトが以下のように送信されます。

OFF: INDOと同じです。さらに、データ・ブロック・タイプ識別子と未定義長ブロック先頭の#0が抑制されます。上のデータは以下のように送信されます。

注: フォーマットOFFはIEEE 488.2標準に適合しません。絶対最小データ転送が重要とされるアプリケーションにのみ適用されます。

データ・タイプ

BYTE（バイト）:波形データを8ビット符号付き整数（1バイト）で送信します。

WORD（ワード）:波形データを16ビット符号付き整数（2バイト）で送信します。

注: BYTEデータ・タイプでは、内部16ビット表現の上位ビットのみ送信されます。下位ビットの精度は失われます。

符号化

BIN: バイナリ符号化（GPIBのみ）

HEX: 16進数符号化（バイトは、2桁の16進数ASCII数字（0, ...9, A ...F）に変換されます）。

関連コマンド

WAVEFORM

通　信　　COMM_HEADER、CHDR

コマンド / クエリ

説　明

COMM_HEADERコマンドは、オシロスコープがクエリに対する応答をフォーマット化する方式を制御します。オシロスコープは次の3種のフォーマットで応答できます。長いフォーマットのヘッダ・ワードで始まるLONGフォーマット、短いフォーマットのヘッダ・ワードで始まるSHORTフォーマット、ヘッダが省略され数値の末尾に単位をつけないOFFです。ユーザが特に指定しない限り、SHORT応答フォーマットが使用されます。

このコマンドは、オシロスコープへ送信されるメッセージの解釈には影響を与えません。ヘッダは、COMM_HEADER設定にかかわらず、長いフォーマットでも短いフォーマットでも送信できます。

チャネル1の垂直感度クエリは、次の3応答のどれかになります。

COMM_HEADER	応　　答
LONG	C1:VOLT_DIV 200E-3V
SHORT	C1:VDIV 200E-3V
OFF	200E-3

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

Comm_HeaDeR <mode>

<mode>:={SHORT, LONG, OFF}

注: デフォルト・モード、つまり電源投入直後はSHORTです。

クエリ構文

Comm_HeaDeR?

応答フォーマット

Comm_HeaDeR <mode>

例（GPIB）

次のコードは、応答ヘッダ・フォーマットをSHORTに設定します。

```
CMD$="CHDR SHORT":CALL IBWRT (SCOPE%, CMD$)
```

説　明

COMM_HELP コマンドはリモート・コントロール・アシスタントを有効にし、リモート・コントロール・プログラムのデバッグを支援します。このリモート・コントロール・アシスタントをオンにすれば、外部のコントローラとwaverunner間で行われている通信を記録することができます。

記録されたログはwaverunnerのスクリーン上で見ることができ、以下の4つのレベルを選択することができます。

OFF	ログを残しません。
EO	エラーが起きたときのみ記録しスクリーンに表示する(電源ON時のデフォルト設定)。
FD	waverunner とコントローラ間の通信をすべて記録しスクリーンに表示します。
RS	waverunner とコントローラ間の通信をすべ記録し、スクリーンのほかにRS-232Cポートにも出力します。

コマンド構文

Comm_HeLP <level>

<level> : = { OFF, EO, FD, RS, }

初期(電源投入後)、<target>はEOです。

クエリ構文

Comm_HeLP?

応答フォーマット

Comm_HeLP <level>

例（GPIB）

次のコードは、リモート・コントロール・アシスタントをONにし、すべての通信を記録し、スクリーンに表示します。

```
CMD$="CHLP FD": CALL IBWRT(SCOPE%,CMD$)
```

関連コマンド

COMM_HELP_LOG

通　信　　COMM_HELP_LOG?、CHL?

クエリ

説　明

COMM_HELP_LOG？クエリは、リモート･コントロール･アシスタント（CHLPの説明を参照）によって発生したログの現在の内容を返します。 もし、パラメータのCLR（オプション）が指定されているならば、ログは伝送された後、クリアされます。CLRが指定されていない時は、ログはクリアされません。

クエリ構文

Comm_HeLP_Log? [CLR]

応答フォーマット

Comm_Help_Log <string containing the logged text>

例（GPIB）

次の命令は、リモート･コントロール・ログを読み出して、それをプリントアウトします。

```
CMD$="CHL?":CALL IBWRT(SCOPE%,CMD$)PRINT
```

関連コマンド

COMM_HELP

説　明

COMM_NETコマンドは、オシロスコープのIPネットワーク・アドレスを指定します。また、LeCroy DSOプリント・ゲートウェイを実行するワークステーションまたはプリンタのネットワーク・アドレスを指定します。COMM_NET?クエリは、現在のネットワーク・アドレスを返します。

コマンド構文

COmm_NET <subaddress>
<subaddress>:={IP,"X.X.X.X.",MASK,"X.X.X.X",GATEWAY,"X.X.X.X",PRINTSVR,"X.X.X.X"}

クエリ構文

COmm_NET?

応答フォーマット

COmm_NET <subaddress>

警　告

このコマンドはすぐに動作します。それ以後のコマンドは変更後のアドレスに対して送るように送信側のソフトウェアを新しいアドレスで初期化する必要があります。

クエリと応答

CONET?
IP,"172.28.11.49",MASK,"255.255.248.0",GATEWAY,"172.28.8.1",PRINTSVR,"172.29.14.68"

次のコマンドは、IPを変更します。
COMM_NET IP,"172.28.11.77"

通　信　　COMM_ORDER、CORD

コマンド / クエリ

説　明

COMM_ORDERコマンドは、波形データ転送のバイト・オーダを制御します。波形データは、最上位バイト（MSB: most significant byte）または最下位バイト（LSB: least significant byte）を開始位置として送信できます。デフォルト・モードではMSBからです。

COMM_ORDERは、波形の記述子とタイム・ブロックに等しく適用されます。記述子には、16ビット長（"word"）、32ビット長（"long"、"float"）、64ビット長（"double"）の値があります。"COMM_ORDER HI"を選択すると、最上位バイトが最初に送信されます。"COMM_ORDER LO"を選択すると、最下位バイトが最初に送信されます。

COMM_ORDER?クエリは、現在使用中のバイト転送順序で返します。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

Comm_ORDer <mode>

<mode>:={HI, LO}

注: 初期モード、つまり電源投入直後はHIです。

クエリ構文

Comm_ORDer?

応答フォーマット

Comm_ORDer <mode>

例

波形データ転送の順番はデータ・タイプにより異なります。次の表に考えられる順序をまとめます。

タイプ	CORD HI	CORD LO
Word	<MSB><LSB>	<LSB><MSB>
Long/Float	<MSB><byte2><byte3><LSB>	<LSB><byte3><byte2><MSB>
Double	<MSB><byte2>...<byte7><LSB>	<LSB><byte7>...<byte2><MSB>

波形データ転送順序

関連コマンド

WAVEFORM

説　明

COMM_RS232コマンドは、RS-232-Cポートのリモート・コントロール・パラメータを設定します。

COMM_RS232?クエリは、パラメータ設定をレポートします。

注。このコマンドは、オシロスコープをRS-232-Cポートを介してリモート・コントロールしている場合に限り有効です。

以下のパラメータがあります。

a. 入力終了（End Input）文字。オシロスコープにこの文字が受信されると、コマンド終了（End-of a command）メッセージ・マーカと解釈されます。受信されたコマンドは、構文解析され実行されます。

b. 出力終了（End Output）文字列。この文字列は、オシロスコープにより応答メッセージの最後に添付されます。ホスト・コンピュータではこの文字列の受信により、オシロスコープの応答完了を認識します。

c. 行長（Line Length）。このパラメータはホストに送信する1行の最大文字数を定義します。応答の残りの文字は、次の行に出力されます。このパラメータは、行区切り文字が選択されている場合にのみ適用されます。

d. 行区切り文字（Line Separator）。このパラメータは、行分割メカニズムを選択し、オシロスコープ応答メッセージを何行にも分けるために使用する文字を定義します。使用できる行区切り文字は、CR、LF、CRLF、<CR>、<LF>、<CR>とその後に続く<LF>です。これらの文字は<line_length>文字の後にホスト・コンピュータに送信されます。

e. SRQ文字列。この文字列は、オシロスコープよりホスト・コンピュータへSRQが発信されるたびに送信されます。

このコマンドのパラメータには、実際の引数にASCII文字列を必要とするものもあります。このような文字列に非印字文字を埋め込むには、エスケープ・シーケンスを使用します。エスケープ文字にはバック・スラッシュ（ 注:日本語の場合はYEN（\）マークになります）を使用します。以下のエスケープ・シーケンスが認識されます。

"＼a":　Bell（ベル）文字
"＼b":　バック・スペース文字
"＼e":　エスケープ文字
"＼n":　改行文字
"＼r":　キャリッジ・リターン文字
"＼t":　水平タブ文字
"＼＼": バック・スラッシュ文字自体
"＼ddd": dddは、対応するASCII文字のコード値を示す1～3桁の10進数を表します。

文字列が使用される前に、オシロスコープではエスケープ文字が対応するASCII文字に置き換えられます。

たとえば、エスケープ・シーケンス"＼r"、"＼13"、"＼013"はすべて1つのASCII文字<Carriage Return（キャリッジ・リターン）>に置き換えられます。

表　　記	
EI	入力終了(End Input)文字
EO	出力終了(End Output)文字列
LL	行長(Line Length)
LS	行区切り文字(Line Separator)
SRQ	SRQサービス・リクエスト

コマンド構文

COmm_RS232 EI, <ei_char>, EO, '<eo_string>', LL, <line_length>, LS, <Line_set>, SRQ, '<srq_string>'

<ei_char>:　　= 1～126（デフォルト: 13 = キャリッジ・リターン）

<eo_string>:　= 空でないASCII文字列20文字まで（デフォルト: "＼n＼r"）

<line_length>: = 40～1024（デフォルト: 256）

<line_sep>:　= {OFF, CR, LF, CRLF}（デフォルト: OFF）

<srq_string>: = 空文字列を含むASCII文字列20文字まで（デフォルト: 空文字列）

クエリ構文

COmm_RS232?

応答フォーマット

COmm_RS232 EI, <ei_char>, EO, "<eo_string>", LL, <line_length>, LS, <line_sep>, SRQ, "<srq_string>"

例

次のコマンド実行後

COMM_RS232 EI, 3, EO, "\r\nEND\r\n"

オシロスコープでは、<ETX>（10進数の3）が検出されるたびに完全な1メッセージが受信されたと見なします。応答メッセージは、文字シーケンス"<CR><LF>END<CR><LF>"送信で終端されます。

捕　捉

COUPLING、CPL

コマンド / クエリ

説　明

COUPLINGコマンドは、指定入力チャネルのカップリング・モードを選択します。

COUPLING?クエリは、指定チャネルのカップリング・モードを返します。

コマンド構文

<channel>:CouPLing <coupling>

<channel>:={C1, C2, C3☝, C4☝, EX, EX10}

<coupling>:={A1M, D1M, D50, GND}

クエリ構文

<channel>:CouPLing?

応答フォーマット

<channel>:CouPLing <coupling>

<coupling>:={A1M, D1M, D50, GND, OVL}

<coupling>: OVLは、DC 50Ωカップリング時に信号過負荷入力で返されます。この状態では、オシロスコープは入力が切り離されます。

☝使用可能対象

<channel>:C3およびC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、チャネル2のカップリングをDC 50Ωに設定します。

```
CMD$="C2:CPL D50":CALL IBWRT (SCOPE%, CMD$)
```

カーソル　CURSOR_MEASURE、CRMS

コマンド / クエリ

説　明

CURSOR_MEASUREコマンドは、表示するカーソルまたはパラメータ測定のタイプを指定します。
CURSOR_MEASURE?クエリは、現在表示されているカーソルまたはパラメータ測定を示します。

表　　記	
ABS	相対カーソルの絶対測定値
CUST	カスタム・パラメータ
FAIL	合否: 不合格
HABS	水平絶対カーソル
HPAR	標準時間パラメータ
HREL	水平相対カーソル
OFF	カーソルおよびパラメータのオフ
PARAM	VPAR と同義語
PASS	合否: 合格
SHOW	カスタム・パラメータ（旧形式）
STAT	パラメータ統計
VABS	垂直絶対カーソル
VPAR	標準電圧パラメータ
VREL	垂直相対カーソル

注: PARAMモードは、XYモードがONのときOFFになります。

コマンド構文

CuRsor_MeaSure <mode>[,<submode>]
<mode>:={CUST, FAIL, HABS, HPAR, HREL, OFF, PARAM, PASS, SHOW, VABS, VPAR, VREL}
<submode>:={STAT, ABS}
注1: キーワードSTATは、CUST、HPAR、VPARモードのオプションです。STATがある場合はパラメータ統計がオンに、ない場合はオフになります。
注2: キーワードABSはHRELモードのオプションです。ABSがある場合は相対カーソルの絶対振幅測定値が、ない場合には相対振幅測定値が選択されます。

クエリ構文

CuRsor_MeaSure?

応答フォーマット

CuRsor_MeaSure <mode>

例（GPIB）

次のコマンドは、垂直相対カーソルを切り替えます。

```
CMD$="CRMS VREL":CALL IBWRT (SCOPE%, CMD$)
```

次のコマンドは、現在オンになっているカーソルを求めます。

```
CMDS$="CRMS?":CALL IBWRT (SCOPE%, CMD$)
CALL IBRD (SCOPE%, RD$) :PRINT RD$
```

応答メッセージ例を以下に示します。

```
CRMS OFF
```

関連コマンド

CURSOR_SET, PARAMETER_STATISTICS,
PARAMETER_VALUE, PASS_FAIL_CLEAR,
PASS_FAIL_CONDITION, PASS_FAIL_DELETE,
PASS_FAIL_MASK,

補足情報

カーソル、パラメータ測定、または合否テストをオフにするには、以下の形式を使用します。

```
CURSOR_MEASURE OFF
```

カーソル表示をオンにするには、以下の4形式のうちいずれか1つを使用します。

```
CURSOR_MEASURE  HABS
CURSOR_MEASURE  HREL
CURSOR_MEASURE  VABS
CURSOR_MEASURE  VREL
```

統計をとらずにパラメータ測定をオンにするには、以下の3形式のうちいずれか1つを使用します。

```
CURSOR_MEASURE  CUST
CURSOR_MEASURE  HPAR
CURSOR_MEASURE  VPAR
```

パラメータ統計をオンにするには、上記の3形式にキーワードSTATを追加します。

パラメータの合否（Pass or Fail）テスト、またはマスク・テストをオンにするには、以下の形式を使用します。

```
CURSOR_MEASURE  PASS
CURSOR_MEASURE  FAIL
```

カスタム（Custom）モードのパラメータを選択し、合否（Pass/Fail）モードのテスト条件を変更するには、以下のコマンドを用います。

```
PASS_FAIL_CONDITION
```

カーソル　CURSOR_READOUT、CRRD?

コマンド / クエリ

説　明

このコマンドは、タイム・カーソルの振幅読出し単位(VまたはdBm)を設定します。

コマンド構文

CuRsor_ReaDout <scale>

<scale>:={VOLTS,DBM}

クエリ構文

CRRD?

応答フォーマット

CURSOR_READOUT VOLTS

例（GPIB）

次のコマンドは、タイム・カーソルの振幅をdBm単位で読み出すよう設定します。

```
CMD$="CRRD DBM"
CALL IBWRT(SCOPE%,CMD$)
```

関連コマンド

CURSOR_MEASURE

カーソル　　CURSOR_SET、CRST?

コマンド / クエリ

説　明

CURSOR_SETコマンドにより、ユーザは、8つのカーソルのうち任意の1カーソルを指定の画面位置に置くことができます。カーソル位置は、目的のカーソルが現在画面に表示されていなくても変更でき、照会できます。

カーソル位置を設定する場合は、そのカーソル位置の基準となるトレースを指定しておく必要があります。

CURSOR_SET?クエリは、複数のカーソルの現在の位置を返します。返される値は、選択したグリッド・タイプにより異なります。

注:パラメータ表示（あるいは合否表示や拡張パラメータ表示）がオンになっていると、新規選択トレースが表示されない場合やシーケンスモードで捕捉完了しなかった場合（これらは環境エラーを引き起こします（ページSC-75を参照してください））を除き、指定トレースのパラメータが表示されます。カーソル位置を変更せずにトレースのみを変更するには、引数なしのCURSOR_SETコマンド（たとえば、.TB:CRST）を使用します。

表　記			
HABS	水平絶対	PREF	パラメータ基準
HDIF	水平差分	VABS	垂直絶対
HREF	水平基準	VDIF	垂直差分
PDIF	パラメータ差分	VREF	垂直基準

コマンド構文

<trace>:CuRsor_SeT <cursor>, <position>[,<cursor>,<position>,...
<cursor>,<position>]
<trace>:={TA, TB, TC, TD, C1, C2, C3☝, C4☝}
<cursor>:={HABS, VABS, HREF, HDIF, VREF, VDIF, PREF, PDIF}
<position>:= 0～10 DIV（水平）
<position>:= -29.5～29.5 DIV（垂直）

注1: 接尾文字DIVはオプションです。

注2: パラメータはペアにグループ化されています。ペアの最初のパラメータは変更するカーソルを指定し、2番目のパラメータで新しい値を示します。ペアは任意の順番に並べることも、変更する変数だけにとどめることもできます。

クエリ構文

<trace>:CuRsor_SeT?[<cursor>,...<cursor>]

<cursor>:={HABS, VABS, HREF, HDIF, VREF, VDIF, PREF, PDIF, ALL}

応答フォーマット

<trace>:CuRsor_SeT<cursore>,<position>[,<curosor>,<position>,... <cursor>,<position>]

<cursor>を指定しない場合は、ALLを指定したと見なされます。なんらかの状況で位置を決定できない場合は、カーソル位置はUNDEFと示されます。

☝使用可能対象

<trace>:C3およびC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、トレースAを基準としてVREFカーソルとVDIFカーソルをそれぞれ+3 DIVと-7 DIVの位置に置きます。

```
CMD$="TA:CRST VREF, 3DIV, VDIF, -7DIV"
CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

CURSOR_MEASURE, CURSOR VALUE, PARAMETER_VALUE,
PER_CURSOR_SET, XY_CURSOR_SET

説　明

CURSOR_VALUE?クエリは、あるトレースにおいて指定カーソルで測定した値を返します（PARAMETER_VALUE?クエリは、測定波形のパラメータ値を返します）。

表　記			
HABS	水平絶対	VABS	垂直絶対
HREL	水平相対	VREL	垂直相対

クエリ構文

<trace>:CuRsor_VAlue?[<mode>,...<mode>]

<trace>:={TA, TB, TC, TD, C1, C2, C3☝, C4☝}

<mode>:={HABS, HREL, VABS, VREL, ALL}

応答フォーマット

<trace>:CuRsor_VAlue HABS,<abs_hori>,<abs_vert>

<trace>:CuRsor_VAlue HREL,<delta_hori>,<delta_vert>,<absvert>,<absvert_dif>

<trace>:CuRsor_VAlue VABS,<abs_vert>

<trace>:CuRsor_VAlue VREL,<delta_vert>

水平カーソルでは水平値も垂直値も与えられます。垂直カーソルでは垂直値のみ与えられます。

注:<mode>指定がない、またはALLの場合は、指定トレースに関する測定カーソル値すべてが返されます。カレント環境でカーソル値を決定できない場合は、値UNDEFが返されます。

☝使用可能対象

<trace>:C3およびC4は、4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のクエリは、チャネル2のクロス・ヘア・カーソル（HABS）の測定絶対水平値を読み取ります。

```
CMD$="C2:CRVA? HABS":CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD (SCOPE%,RSP$) :PRINT RSP$
```

応答メッセージ:

C2:CRVA HABS, 34.2E-6 S, 244 E-3V

関連コマンド

CURSOR_SET, PARAMETER_VALUE, PER_CURSOR_VALUE,
XY_CURSOR_VALUE

SC-46::42.1::CURSOR_VALUE, CRVA, Query【CURSOR_VALUE、CRVA、クエリ】

説　明

DATA_POINTSコマンドは、波形サンプル・ポイントをシングル表示ピクセルで表示するか太字表示にするか制御します。

DATA_POINTS?クエリの応答は、波形サンプル・ポイントがシングル表示ピクセルで表示されているか太字で表示されているかを返します。

コマンド構文

Data_PoiNTs <state>

<state>:={NORMAL, BOLD}

クエリ構文

Data_PoiNTs?

応答フォーマット

Data_PoiNTs <state>

例（GPIB）

次の命令は、波形サンプル・ポイントをハイライトします。

```
CMD$="DPNT ON":CALL IBWRT (SCOPE%, CMD$)
```

その他

DATE

コマンド / クエリ

説　明

DATEコマンドは、オシロスコープの内部リアル・タイム・クロックの日時を変更します。

DATE?クエリは、カレント日時設定を返します。

コマンド構文

DATE <day>,<month>,<year>,<hour>,<minute>,<second>

<day>:=1～31

<month>:={JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC}

<year>:=1990～2037

<hour>:=0～23

<minute>:=0～59

<second>:=0～59

注:dateパラメータすべてを常に指定する必要はありません。更新するパラメータのみ、および変更するパラメータを含むもののみ指定します。"year（年）"の設定を変更するには、day（日）、month（月）、year（年）を一緒に設定します。時間設定は無変更のままです。"second（秒）"設定を変更するには、DATEパラメータすべてを設定する必要があります。

クエリ構文

DATE?

応答フォーマット

DATE <day>,<monthe>,<year>, <hour>, <minute>, <second>

例（GPIB）

この例は、日付を1997年1月1日に、時刻を1:21:16 p.m.（24時間表示では13:21:16）に変更します。

```
CMD$="DATE 1, JUN, 1997, 13, 21, 16":
CALL IBWRT (SCOPE%, CMD$)
```

ステータス

DDR?

クエリ

説　明

DDR?は、デバイス依存エラー・レジスタまたはデバイス特定エラー・レジスタ（DDR: Device Dependent or device specific error Register）の読み出しとクリアを行います。ハードウェア障害の場合は、DDRレジスタに障害発生源が指定されます。以下の表に詳細をまとめます。

ﾋﾞｯﾄ	ﾋﾞｯﾄ値	説　　明
15...14		0　予約
13	8192	1　タイムベース・ハードウェア障害検出
12	4096	1　トリガ・ハードウェア障害検出
11	2048	1　チャネル4* ハードウェア障害検出
10	1024	1　チャネル3* ハードウェア障害検出
9	512	1　チャネル2 ハードウェア障害検出
8	256	1　チャネル1 ハードウェア障害検出
7	128	EXT TRIG オーバロード状態検出
6...4		0　予約
3	8	1　チャネル4* 過負荷状態検出
2	4	1　チャネル3* 過負荷状態検出
1	2	1　チャネル2 過負荷状態検出
0	1	1　チャネル1 過負荷状態検出

デバイス特定レジスタ構造（DDR）

クエリ構文

DDR?

応答フォーマット

DDR <value>

<value>:= 0～65535

☝使用可能対象

<value>:Bit2、3、10、11－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次の命令は、DDRレジスタ内容を読み出します。

```
CMD$="DDR":CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
応答メッセージ:
DDR 0
```

関連コマンド

ALL_STATUS, *CLS

説　明

DEFINEコマンドは、演算で使用する関数をコントロールします。使用できる演算の種類は搭載されているオプションによって異なりますのでご注意ください。

コマンド構文

<function> : DEFine EQN,"<equation>" [,<param_name>,<value>,...]

注: このコマンドで指定されるパラメータは、必ずペアになっています。例えば、MAXPTSというパラメータには<max_points>というペアになるパラメータが必要です。パラメータはペア単位で任意にグループ化でき、変更する変数だけにとどめることもできます。

クエリ構文

<function> : DEFine?

応答フォーマット

<function> : DEFine EQN,"<equation>"[,MAXPTS,<max_points>] [,SWEEPS,<max_sweeps>] [,WEIGHT,<weight>][,BITS,<bits>]...

標準でサポートされるパラメーター一覧		
<param_name>	<value>	説明
BITS	<bits>	ERES ビット数
EQN	"<equation>"	演算名称(次ページの表参照)
MAXPTS	<max_points>	演算に使用する最大ポイント数
SWEEPS	<max_sweeps>	最大スイープ数
UNITS	<units>	リスケール演算の工学単位
WINDOW	<window_type>	FFT の窓関数の種類
EMM で追加されるパラメーター一覧		
WEIGHT	<weight>	連続アベレージの重み
WAVA で追加されるパラメーター一覧		
CENTER		ヒストグラム表示の水平中央位置
MAX_EVENTS	<max_values>	ヒストグラムの値の最大値
MAXBINS	<bins>	ヒストグラムのビンの値
VERT	<vert_scale>	ヒストグラムの垂直軸スケールタイプ
WIDTH	<width>	ヒストグラムの表示幅

標準でサポートされる関数一覧	
<function>	説　　明
((<source>	否定
+<source>	同定
<source>	同定
<source1> - <source2>	減算
<source1> + <source2>	加算
<source1>/<source2>	除算
<source1><source2>	乗算
AVGS(<source>)	加算アベレージ
ERES(<source>)	分解能向上
EXTR(<source>)	エクストレーマ(エンベロープ)
FLOOR(EXTR(<source>))	エクストレーマのフロア
RESC([{+,(}][<multiplier>]*<source>	リスケール
[{+,(}<addend>])	
ROOF(EXTR(<source>))	エクストレーマのルーフ
SINX(<source>)	サイン補間
ZOOMONLY (<extended_source>)	ズームのみ(演算無し)
MAG(FFT(<source>))	FFT 演算結果の振幅部
PHASE(FFT(<source>))	FFT 演算結果の位相部
PS(FFT(<source>))	パワー・スペクトラム
EMM で追加される関数一覧	
1/<source>	逆数
ABS(<source>)	絶対値
AVGC(<source>)	連続アベレージ
DERI(<source>)	微分
EXP(<source>)	e のべき乗
EXP10(<source>)	10 のべき乗
INTG(<source>[{+,(} <addend>])	積分
LN(<source>)	自然対数
LOG10(<source>)	常用対数
SQR(<source>)	二乗
SQRT(<source>)	平方根

WAVA で追加される関数一覧	
FFT(<source>)	高速フーリエ変換(複素数解)
IMAG(FFT(<source>))	FFT 演算・複素数解の虚数部
REAL(FFT(<source>))	FFT 演算・複素数解の実数部
MAG(AVGP(<function>))	FFT 演算・振幅部のアベレージ
PS(AVGP(<function>))	パワー・スペクトラムのアベレージ
PSD(AVGP(<function>))	パワー密度のアベレージ
HIST(<custom_line>)	指定されたカスタム・パラメータのヒストグラム
TREND(<custom_line>)	指定されたカスタム・パラメータのトレンド

ソース値

<sourceN> : = {TA, TB, TC, TD, M1, M2, M3, M4, C1, C2, C3G, C4G}

<function> : = {TA, TB, TC, TD}

<custom_line> : = {CUST1, CUST2, CUST3, CUST4, CUST5}

<extended_source> : = {C1, C2, C3G,C4G, TA, TB, TC, TD, M1, M2, M3, M4}

ポイント数/スイープ数:

<max_points> : = 50 to 10 000 000

<max sweeps> : = 1 to 1000

<max_sweeps> : = 1 to 1 000 000 (WAVAオプションのみ)

リスケール用の値:

<addend> : = 0.0 to 1e15

<multiplier> : = 0.0 to 1e15

加算平均とERES演算用の値:

<weight> : = {1, 3, 7, 15, 31, 63, 127, 255, 511, 1023}

<bits> : = {0.5, 1.0, 1.5, 2.0, 2.5, 3.0}

FFTの窓関数用の値:

<window_type> : = {BLHA, FLTP, HAMM, HANN, RECT}

FFT の窓関数表記	
BLHA	ブラックマン－ハリス
FLTP	フラット・トップ
HAMM	ハミング
HANN	ハニング
RECT	方形

ヒストグラム用の値:

<max bins> : = {20, 50, 100, 200, 500, 1000, 2000} (WAVAオプションにて有効)
<max_events> : = 20 to 2e9 (in a 1-2-5 sequence)
 : = (1e15 to 1e15
<width> : = 1e(30 to 1e30 (in a 1-2-5 sequence)
<vert_scale> : = {LIN, LOG, CONSTMAX}

ヒストグラムの表記	
LIN	ヒストグラムの表示にリニア・スケールを使用
LOG	ヒストグラムの表示にログ・スケールを使用
CONSTMAX	ヒストグラムの表示に定最大リニア・スケールを使用

使用可能対象

<sourceN> : = {C3, C4} は4チャンネル・モデルのみ
<extended_source> : = {C3, C4} は4チャンネル・モデルのみ

例（GPIB）

次のコマンドは、トレースAを5000ポイント使用して、チャンネル1の波形を200回加算アベレージします。

```
CMD$="TA:DEF EQN,"AVGS(C1)",MAXPTS,5000,SWEEPS,200":
CALL IBWRT(SCOPE%,CMD$)
```

次のコマンドは、トレースAを10000ポイント使ってチャンネル1とチャンネル2の乗算を実行します。

```
CMD$="TA:DEF EQN,"C1*C2",MAXPTS,10000":
CALL IBWRT(SCOPE%,CMD$)
```

以下のコマンドは、トレースAを1000ポイント使ってチャンネル1のパワー・スペクトラムを計算します。窓関数としては、方形窓を指定しています。

```
CMD$="TA:DEF EQN,"PS(FFT(C1))",MAXPTS,1000,WINDOW, RECT":
CALL IBWRT(SCOPE%,CMD$)
```

以下のコマンドは、トレースBにトレースAで行われているパワー・スペクトラム波形を244回アベレージして表示します。

```
CMD$="TB:DEF EQN,"PS(AVGP(TA))",SWEEPS,244": CALL IBWRT(SCOPE%,CMD$)
```

以下のコマンドは、トレースCに、カスタム・ライン2行目で行われているチャンネル1の立上がり時間の値をヒストグラムとして表示します。ヒストグラムはリニア・スケールを持ち、ビンの値は100です。

```
CMD$="PACU 2,RISE"
CALL IBWRT(SCOPE%,CMD$)
CMD$="TC:DEF EQN,"HIST(CUST2)",VERT,LIN,MAXBINS,100"
CALL IBWRT(SCOPE%,CMD$)
```

関連コマンド

FIND_CTR_RANGE, FUNCTION_RESET, INR?, PARAMETER_CUSTOM, PARAMETER_VALUE?, PASS_FAIL_CONDITION

保存装置　　DELETE_FILE、DELF コマンド

説　明

DELETE_FILEコマンドは、保存装置内の現在選択されているディレクトリからファイルを削除します。

コマンド構文

DELete_File DISK, <device>, FILE,'<filename>'

<device>:={CARD☝, FLPY, HDD☝}

<filename>:= 8文字までの英数文字列に、ドットと3文字までの拡張子が続きます。

☝使用可能対象

<device>:CARD－メモリカードオプション装着時にのみ使用できます。

<device>:HDD－ハードディスクオプション装着時にのみ使用できます。

例（GPIB）

次のコマンドは、フロント・パネル・セットアップをメモリ・カードより削除します。

```
CMD$="DELF DISK, CARD, FILE,'P001.PNL'":
CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

DIRECTORY, FORMAT_CARD, FORMAT_FLOPPY,
FORMAT_HDD

保存装置

DIRECTORY、DIR

コマンド / クエリ

説　明

DIRECTORYコマンドは、保存装置のファイル・ディレクトリの作成と削除を管理します。また、カレント作業ディレクトリの選択とディレクトリ内のファイルのリスト表示も行えます。

クエリの応答は、DOSに似たディレクトリ・リストの文字列が二重引用符で囲まれて構成されます。保存装置がない、またはフォーマットされていない場合には、文字列は空です。

コマンド構文

DIRectory DISK, <device>, ACTION, <action>,'<directory>'

クエリ構文

DIRectory? DISK, <device>[,'<directory>']

<device>:={CARD☝, FLPY, HDD☝}

<action>:={CREATE, DELETE, SWITCH}

<directory>:=正規のDOSパス名またはファイル名（これには、'\'文字を入れてルート・ディレクトリを定義できます）。

注: DIRectory_list?クエリは、下位互換性のために受け付けられますが、将来はサポートされない可能性があります。

応答フォーマット

DIRectory DISK, <device>"<directory>"

<directory>:=メモリ・カード、フロッピー・ディスク、またはハード・ディスクのファイル内容詳細を示す可変長文字列です。

☝使用可能対象

<device>:CARD－メモリカードオプション装着時にのみ使用できます。

<device>:HDD－ハードディスクオプション装着時にのみ使用できます。

例（GPIB）

次のコードは、メモリ・カードのディレクトリ一覧を要求します。

```
CMD$="DIR? DISK, CARD":CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
応答メッセージ
DIR"
Directory   LECROY      1 DIR of  04-MAR-1994     10:4620 on Memory Card
SC1         000           2859    19-DEC-1994     16:33:06
SC1         001           2859    19-DEC-1994     16:34:32
TEST5       002         20359     12-MAR-1994     13:34:12
3 File (s)  1948672 bytes free
"
```

表　示

DISPLAY、DISP

コマンド / クエリ

説　明

DISPLAYコマンドは、オシロスコープの表示画面を制御します。ユーザがオシロスコープをリモート・コントロールしていて、オシロスコープの表示が要らない場合に、DISPLAY OFFコマンドで表示をオフにするときに使用します。これにより波形グラフィック生成手続きを抑止してオシロスコープの応答時間を短縮できます。

DISPLAY?クエリに対する応答は、オシロスコープの表示ステートを返します。

注:表示がOFFに設定されているとき、リアル・タイム・クロックとメッセージ・フィールドは更新されますが、波形と関連テキストは未変更のままです。

コマンド構文

DISPlay <state>

<state>:={ON, OFF}

クエリ構文

DISPlay?

応答フォーマット

DISPlay <state>

例（GPIB）

次の命令は、表示生成をオフにします。

```
CMD$=:"DISP OFF":CALL IBWRT (SCOPE%,CMD$)
```

表　示

DOT_JOIN、DTJN

コマンド / クエリ

説　明

DOT_JOINコマンドは、データ・ポイント間の補間線を制御します。

コマンド構文

DoT_JoiN <state>

<state>:={ON, OFF}

クエリ構文

DoT_JoiN?

応答フォーマット

DoT_JoiN <state>

例（GPIB）

次の命令は、補間線をオフにします。

CMD$="DTJN OFF":CALL IBWRT（SCOPE%, CMD$）

DUAL_ZOOM、DZOM

コマンド / クエリ

説　明

DUAL_ZOOMコマンドの設定により、すべての拡張トレースの水平拡大率と位置設定を同時に制御できます。このコマンドは、すべての拡張トレース内容を1度に調べる時に便利です。

DUAL_ZOOM?クエリは、マルチプル・ズームがイネーブルかディスエーブルかを返します。

注: このコマンドでは、MULTI_ZOOMと同じ効果が得られます。

コマンド構文

Dual_ZOoM <mode>

<mode>:={ON, OFF}

クエリ構文

Dual_ZOoM?

応答フォーマット

Dual_ZOoM <mode>

例（GPIB）

次の例は、デュアル・ズームをオンにします。

```
CMD$="DZOM ON":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

HOR_MAGNIFY, HOR_POSITION, MULTI_ZOOM

ステータス

ENABLE_KEY、EKEY

コマンド / クエリ

説　明

ENABLE_KEY コマンドは、メニューをローカル・モードで下の6個のラベル付きメニューボタンに割り当てることができます。

コマンド構文

ENABLE_KEY <state>

<state>: ={ON, OFF}

クエリ構文

ENABLE_KEY?

応答フォーマット

ENABLE_KEY <state>

例（GPIB）

```
CMD$="ENABLE_KEY":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

KEY

ステータス　　*ESE

コマンド / クエリ

説　明

*ESEコマンドは、標準イベント・ステータス・イネーブル（ESE: Standard Event Status Enable）レジスタを設定します。このコマンドにより、ESRレジスタの複数のイベントをSTBレジスタのESBサマリ・メッセージ・ビット（ビット5）に反映できます。ESB定義イベントの概略は、ページSC-72のESR表を参照してください。

*ESE?クエリは、ESEレジスタの内容を読み出します。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

*ESE <value>

<value>:=0～255

クエリ構文

*ESE?

応答フォーマット

*ESE <value>

例（GPIB）

次のコマンドにより、ユーザ要求（URQビット6、つまり10進数の64）および/またはデバイス依存エラー（DDEビット3、つまり10進数の8）が発生したときに、ESBビットを設定することができます。これらの値の和は、ESEレジスタ・マスク64 + 8 = 72となります。

```
CMD$="*ESE 72":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

*ESR

ステータス　*ESR?

クエリ

説　明

*ESR?クエリは、イベント・ステータス・レジスタ（ESR: Event Status Register）の読み出しとクリアを行います。このクエリに対する応答は、レジスタビット0と7のバイナリ値の和を返します。下の表に、ESRレジスタ構造の概略を示します。

クエリ構文

*ESR?

応答フォーマット

*ESR <value>

<value>:= 0～255

例（GPIB）

次の命令は、ESRレジスタの内容を読み出し、クリアします。

CMD$="*ESR?":CALL IBWRT (SCOPE%, CMD$) :

CALL IBRD (SCOPE%, RSP$) :PRINT RSP$

応答メッセージ:

*ESR 0

関連コマンド

ALL_STATUS, *CLS, *ESE

補足情報

ﾋﾞｯﾄ	ﾋﾞｯﾄ値	ﾋﾞｯﾄ名	説　明	注
15...8			IEEE 488.2で予約	
7	128	PON	電源オフからオン(Power off-to-ON)遷移発生	(1)
6	64	URQ	ユーザ・リクエスト(User ReQuest)が発行された	(2)
5	32	CME	コマンド解析エラー(CoMmand parser Error)が検出された	(3)
4	16	EXE	実行エラー(EXecution Error)が検出された	(4)
3	8	DDE	デバイス特定エラー(Device specific Error)発生	(5)
2	4	QYE	クエリ・エラー(QuerY Error)発生	(6)
1	2	RQC	オシロスコープよりバス制御が要求されていない	(7)
0	1	OPC	動作完了(OPeration Complete)ビット未使用	(8)

標準イベント・ステータス・レジスタ（ESR: Standard Event Status Register）

注

(1) Power ON (PON) ビットは、ユニットに電源が投入されるとオン (1) になります。

(2) ユーザリクエスト (URQ: User ReQuest) ビットは、ソフト・キーが押されるとTRUE (1) になります。関連レジスタURRは、選択されたキーを特定します。詳細は、URR?クエリを参照してください。

(3) コマンド解析エラー (CME: CoMmand parser Error) ビットは、コマンド構文エラーが検出されるとTRUE (1) になります。CMEビットには、関連コマンド・パーサー・レジスタ (CMR: CoMmand parser Register) があり、これによってエラー・コードを示します。詳細は、CMR?クエリを参照してください。

(4) 実行エラー (EXE: EXecution Error) ビットは、何らかのデバイス状態 (たとえばオシロスコープがローカル・ステート) であったり意味内容にエラーがあって、コマンド実行できない場合にTRUE (1) になります。EXEビットには、関連実行エラー・レジスタ (EXR: Execution Error Register) があり、これによってエラー・コードを示します。詳細は、EXR?クエリを参照してください。

(5) デバイス特定エラー (DDE: Device specific Error) ビットは、電源投入時に、またはチャネル過負荷状態、トリガまたはタイムベース回路障害など、実行時にハードウェア障害が発生するとTRUE (1) になります。障害発生原因は、DDR?クエリまたは自己試験*TST?クエリにより検出できます。

(6) クエリ・エラー (QYE: QuerY Error) ビットは、次の場合にTRUE (1) になります。(a) 出力がないときに、または保留中に出力待ち行列 (Output Queue) より読み出しが試みられた場合。(b) 出力待ち行列のデータが失われた場合。(c) 出力バッファと入力バッファ両方が満杯 (デッドロック・ステート) の場合。(d) コントローが<END>送信完了前に読み出しを試みた場合。(e) 前のクエリに対する応答が読み出される (出力バッファがフラッシュされる) 前にコマンドが受信された場合。

(7) リクエスト制御 (RQC: ReQuest Control) ビットは、オシロスコープにGPIB制御機能がないため常にFALSE (0) です。

(8) 動作完了 (OPC: OPeration Complete) ビットは、コマンドとクエリは厳密にシーケンシャルな順番で実行されるため、*OPC受信時は常にTRUE (1) になります。オシロスコープは、前のコマンドをすべて実行完了した時にのみ次のコマンドの処理を開始します。

ステータス

EXR?

クエリ

説　明

EXR?クエリは、実行エラー・レジスタ（EXR: EXecution error Register）の読み出しとクリアを行います。EXRレジスタは、実行中最後に検出されたエラーのタイプを示します。詳細は、ページSC-71の表を参照してください。

クエリ構文

EXR?

応答フォーマット

EXR <value>

<value>:= 21～64

例（GPIB）

次の命令は、EXRレジスタの内容を読み出し、クリアします。

```
CMD$="EXR?":CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
応答メッセージ（障害のない場合）:
EXR 0
```

関連コマンド

ALL_STATUS, *CLS

補足情報

値	説　　明
21	許可エラー。コマンドは、ローカル・モードでは実行できません。
22	環境エラー。オシロスコープ構成が不適切でコマンドを処理できません。たとえば、オシロスコープは低速タイムベースではRISを送信できません。
23	オプション・エラー。該当コマンド適用に必要なオプションがインストールされていません。
24	構文解析エラー。
25	パラメータ・エラー。パラメータ指定が多すぎます。
26	未実装コマンド。
30	16進データ・エラー。16進データ・ブロックで16進数でない文字が検出されました。
31	波形エラー。受信データ量が記述子インジケータに対応しません。
32	波形記述子エラー。無効の波形記述子が検出されました。
33	波形テキスト・エラー。壊れた波形ユーザ・テキストが検出されました。
34	波形時間エラー。無効のRISまたはTRIG時間データが検出されました。
35	波形データ・エラー。無効の波形データが検出されました。
36	パネル・セットアップ・エラー。無効のパネル・セットアップ・データ・ブロックが検出されました。
50	ユーザが保存装置へのアクセスを試みたときに保存装置がありませんでした。@
51	ユーザが保存装置へのアクセスを試みたときに保存装置がフォーマットされていませんでした。@
53	ユーザがファイルの作成または削除、またはデバイスのフォーマットを試みたときに保存装置に書き込み保護がかかっていました。@
54	フォーマット中、保存装置不良が検出されました。@
55	保存装置ルート・ディレクトリ満杯。ディレクトリを追加できません。@
56	ユーザが保存装置への書き込みを試みたときに保存装置が満杯でした。@
57	保存装置のファイル・シーケンス番号が溢れています(999に達しました)。@
58	保存装置のファイルが見つかりません。@
59	要求ディレクトリが見つかりません。@
61	保存装置のファイル名が非DOS互換、または不正ファイル名です。@
62	同じファイル名のファイルが既に保存装置にあるので、ファイルを書き込めません。@
@メモリ・カード(MC01)、フロッピー・ディスク(FD01)、またはハード・ディスク(HD01)オプションを装着したオシロスコープのみ。	

実行エラー・ステータス・レジスタ（EXR）の構造

表　示

FAT_CURSOR、FATC

コマンド / クエリ

説　明

FAT_CURSOR で、カーソルの太さを調節します。

コマンド構文

FAT_CURSOR,FATC <state>

<state>: ={ON, OFF}

クエリ構文

FAT_CURSOR?

応答フォーマット

FAT_CURSOR <state>

例（GPIB）

次のコマンドは カーソルを太くするときに用います:

```
CMD$="FAT_CURSOR ON":CALL IBWRT (SCOPE%, CMD$)
```

説　明

FILENAMEコマンドは、保存デバイスへトレース、セットアップ、ハード・コピーなどを保存するときに、これらにデフォルトで割り当てられているファイル名を変更します。

コマンド構文

FiLeNaMe TYPE, <type>, FILE, “<filename>”

<type>:={C1, C2, C3, C4, TA, TB, TC, TD, SETUP, HCOPY}

<filename>: =C1からTDまでは、正規のDOSファイル名形式の8文字までの英数文字列。SETUPとHCOPYには、5文字まで。

注: 拡張子は、オシロスコープにより自動割り当てされるため変更できません。

クエリ構文

FiLeNaMe? TYPE, <type>

<type>:={ALL, C1, C2, C3, TA, TB, TC, TD, SETUP, HCOPY}

応答フォーマット

FiLeNaMe

TYPE, <type>, FILE, "<filename>"[,TYPE, <type>, FILE," <filename>"...]

例（GPIB）

次のコマンドは、チャネル1の波形ファイル名を"TESTPNT6.xxx"に指定します。ここで、xxxはオシロスコープにより割り当てられた数値拡張子です。

```
CMD$="FLNM TYPE, C1, FILE, 'TESTPNT6'":
CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

DIRECTORY, FORMAT_CARD, FORMAT_FLOPPY,

FORMAT_HDD, DELETE_FILE

関　数　　FIND_CTR_RANGE、FCR

コマンド

説　明

FIND_CTR_RANGE コマンドはヒストグラムの中央の位置と幅を設定して、蓄積された現象を最適な状態で表示します。

コマンド構文

<function>:Find_Ctr_Range

<function>: ={TA,TB,TC,TD}

☝使用可能対象

ヒストグラムのオプションをインストール時のみ使用できます。

例（GPIB）

トレースA(TA)がカスタム・パラメータのヒストグラムであると想定した場合、次の例では、ヒストグラムの最適な中央位置と幅を設定し、再度計算します。

関連コマンド

DEFINE, PACU

説　明

FORMAT_CARDコマンドは、PCMCIA/JEIDA標準準拠のメモリ・カードをDOSパーティションを用いてフォーマッティングします。

FORMAT_CARD?クエリは、カードのステータスを返します。

コマンド構文

Format_CaRD

クエリ構文

Format_CaRD?

応答フォーマット

Format_CaRD <card_status>[,<read/write>,<free_space>,<card_size>,<battery_status>]

<card_status>:={NONE, BAD, BLANK, DIR_MISSING, OK}

<read/write>:={WP, RW}

<free_space>:= カードで使用できる空きバイト数を10進数で示します。

<card_size>:= カード容量総バイト数を10進数で示します。

<battery_status>:{BAT_OK, BAT_LOW, BAT_BAD}

☝使用可能対象

メモリカードオプションを装着したオシロスコープでのみ使用できます。

例（GPIB）

次のコードは、まずメモリ・カードをフォーマットし、その後そのステータスを検証します。

```
CMD$="FCRD":CALL IBWRT (SCOPE%, CMD$)
CMD$="FCRD?":CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
```

応答メッセージ：

```
FCRD OK, RW, 130048, 131072, BAT_OK
```

関連コマンド

DIRECTORY

補足情報

表　　記	
BAD	フォーマッティング後カード不良
BAT_BAD	バッテリ不良またはバッテリーなし
BAT_LOW	バッテリー交換必要
BAT_OK	バッテリー正常
BLANK	カレント・ディレクトリが空
DIR_MISSING	サブディレクトリなし。次の "store" コマンドでディレクトリ "LECROY1_DIR" が自動的に作成されます。
NONE	カードなし
OK	カードは適切にフォーマットされています。
RW	読み出し / 書き込み許可
WP	書き込み禁止

説　明

FORMAT_FLOPPYコマンドは、フロッピー・ディスクを倍密（DD: Double Density）または高密(HD: High Density) にフォーマッティングします。

FORMAT_FLOPPY?クエリは、フロッピー・ディスクのステータスを返します。

コマンド構文

Format_FLoPpy [<type>]

<type>:={DD, HD}

引数を指定しないと、デフォルトでHDが指定されます。

クエリ構文

Format_FLoPpy?

応答フォーマット

Format_FLoPpy <floppy_status>[,<read/write>,<free_space>,<floppy_size>]

<floppy_status>:{NONE, BAD, BLANK, DIR_MISSING, OK}

<read/write>:={WP, RW}

<free_space>:=フロッピーで使用できる空きバイト数を10進数で示します。

<floppy_size>:=フロッピー容量総バイト数を10進数で示します。

例（GPIB）

次のコードは、まずフロッピーをDD（720 kB）フォーマットし、その後そのステータスを検証します。

```
CMD$="FFLP DD":IBWRT (SCOPE%, CMD$)
CMD$="FFLP?:CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
```

応答メッセージ：

FFLP OK, RW, 728064, 737280

関連コマンド

DIRECTORY

補足情報

表　　記	
BAD	フォーマッティング後フロッピー不良
BLANK	カレント・ディレクトリが空
DD	DD 720 kB にフォーマット
DIR_MISSING	サブディレクトリなし。次の "store" コマンドでディレクトリ "LECROY1_DIR" が自動的に作成されます。
HD	HD 1.44 MB にフォーマット
NONE	フロッピーなし
OK	フロッピーは適切にフォーマットされています。
RW	読み出し / 書き込み許可
WP	書き込み禁止

説　明

FORMAT_HDDコマンドは、PCMCIA/JEIDA標準準拠の取り外し可能ハード・ディスクをDOSパーティションを用いてフォーマッティングします。

FORMAT_HDD?クエリは、ハード・ディスクのステータスを返します。

コマンド構文

Format_HDD <type>

<type>:={QUICK, FULL}

引数を指定しないと、QUICKが使用されます。

クエリ構文

Format_HDD?

応答フォーマット

Format_HDD <hdd_status>[,<read/write>,<free_space>,<hdd_size>]

<hdd_status>:={NONE, BAD, BLANK, DIR_MISSING, OK}

<read/write>:={WP, RW}

<free_space>:= ハード・ディスクで使用できる空きバイト数を10進数で示します。

<hdd_size>:= ハード・ディスク容量総バイト数を10進数で示します。

《入力省略》

☝使用可能対象

ハードディスクオプションを装着したオシロスコープでのみ使用できます。

例（GPIB）

次のコードは、まずハード・ディスクをフォーマットし、その後そのステータスを検証します。

```
CMD$="FHDD":CALL IBWRT (SCOPE%, CMD$)
CMD$="FHDD?":CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
```

応答メッセージ：

FHDD OK, RW, 3076096, 105744896

関連コマンド

DIRECTORY

補足情報

表　　記	
BAD	フォーマッティング後ハード・ディスク不良
BLANK	カレント・ディレクトリが空
DIR_MISSING	サブディレクトリなし。次の "store" コマンドでディレクトリ "LECROY1_DIR" が自動的に作成されます。
NONE	ハード・ディスクなし
OK	ハード・ディスクは適切にフォーマットされています。
RW	読み出し / 書き込み許可
WP	書き込み禁止

表　示

FULL_SCREEN、FSCR

コマンド / クエリ

説　明

FULL_SCREENコマンドは、現在選択中のグリッド・スタイルを通常プレゼンテーション・フォーマット、またはフル・スクリーン・グリッドのどちらで表示するかを制御します。フル・スクリーン・フォーマットでは、波形表示領域は最大サイズに拡大されます。

FULL_SCREEN?クエリに対する応答は、フルスクリーン・プレゼンテーション・フォーマットで表示されているかどうかを示します。

コマンド構文

FullSCReen <state>

<state>:={ON, OFF}

クエリ構文

FullSCReen?

応答フォーマット

FullSCReen <state>

例（GPIB）

次の命令は、フルスクリーン・プレゼンテーション・フォーマットをイネーブルにします。

CMD$="FSCR ON":CALL IBWRT (SCOPE%, CMD$)

関　数　FUNCTION_RESET、FRST コマンド

説　明

FUNCTION_RESETコマンドは、波形処理関数をリセットします。スイープ数をゼロにリセットし、プロセスをリスタートします。

コマンド構文

<function>:Function_ReSeT

例（GPIB）

<function>:={TA, TB, TC, TD}

トレースA（TA: Trace A）にチャネル1の加算平均が定義されている仮定すると、以下の例は、平均処理をリスタートします。

```
CMD$="TA:FRST":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

DEFINE, INR

説　明

GLOBAL_BWL コマンドは、GBWのON/OFFを切り替えます。

GBWがONのとき、選択された帯域制限はすべてのチャネルに適用します。

また、GBWがOFFのときは、帯域制限は、チャネルごとに設定することができます。 (BWL参照)。 GLOBAL_BWL? クエリに対する応答は GBWがONかOFFのどちらになっているかを返します。

コマンド構文

Global_BWL <mode>

<mode>: ={OFF, ON}

クエリ構文

Global_BWL?

応答フォーマット

Global_BWL <mode>

例（GPIB）

次のコマンドはGWLをOFFにして、チャネルごとに帯域制限を設定できるようにします。（その際、BWLコマンドによって各チャネルの帯域制限を設定します。）

CMD$="GBWL OFF":CALL IBWRT (SCOPE%, CMD$)

関連コマンド

BANDWIDTH_LIMIT

表　示　　GRID

コマンド / クエリ

説　明

GRIDコマンドは、のグリッド表示をシングル・モード、デュアル・モード、またはクワッド・モードのどれにするか指定します。

GRID?クエリは、現在使用中のグリッド・モードを返します。

コマンド構文

GRID <grid>

<grid>:={SINGLE, DUAL, QUAD, OCTAL ☝, XYONY}

クエリ構文

GRID?

応答フォーマット

GRID <grid>

例（GPIB）

次のコマンドは、画面表示をデュアル・グリッド・モードに設定します。

CMD$="GRID DUAL":CALL IBWRT (SCOPE%, CMD$)

XY_DISPLAYコマンドにてXYディスプレーがOFFになっている場合、<gaid>:=XYONLYは受けつけません。

XY_DISPLAYコマンドにてXYディスプレーがONになっている場合、<grid>は、SINGLE, DUAL, R1及びXYONLYが有効になります。

関連コマンド

XY_DISPLAY

ハード・コピー

HARDCOPY_SETUP、HCSU

コマンド / クエリ

説　明

HARDCOPY_SETUPコマンドは、オシロスコープのハード・コピー・ドライバを構成します。このコマンドにより、ユーザは、オシロスコープに接続されているハード・コピー・ユニットのデバイス・タイプと送信モードを指定できます。適切なキーワードと新しい値を複数指定して、個々の設定を複数変更できます。

コマンド構文

HardCopu_SetUp DEV, <device>, PORT, <port>, PFEED,
<page_feed>, PENS, <plot_pens>, PSIZE, <paper_size>.
[,CMDIV,<cmdiv>, AUTO,<auto>, FORMAT,<format>], BCKG <bckg>.
<device>:= {BMP, HPDJ, HPDJBW, HPPJ, HPTJ, HPLJ, HP7470A, HP7550A, EPSON, EPSON COL,
TIFF, TIFFCOMP, TIFFCOL}
<port>:={GPIB, RS, CENT, FLPY, CARD☝, HDD☝, PRT☝}
<page_feed>:={ON, OFF}
<plot_pens>: 1～8☝
<paper_size>:={A5, A4}☝
<cmdiv>:={1, 2, 5, 10, 20, 50, 100, 200}
<auto>:={OFF, ON}
<format>:{PORTRAIT, LANDSCAPE}
<bckg>:={BLACK, WHITE}

クエリ構文

HardCopy_SetUp?

応答フォーマット

HardCopy_SetUp DEV,<device>, PORT,<port>,
PFEED,<page_feed>, PENS,<plot_pens>, PSIZE,<paper_size>
CMDIV,<cmdiv>, AUTO,<auto>, FORMAT,<format>

☝使用可能対象

<card>:CARD - メモリカードオプション装着時のみ使用できます。
<port>:HDD - ハードディスクオプション装着時のみ使用できます。
<port>:PRT - GPO2オプション装着時のみ使用できます。
<cmdiv> - GPO2オプション装着時のみ使用できます。
<auto> - GPO2オプション装着時のみ使用できます。

例（GPIB）

下の例は、RS232ポートのEPSONプリンタ接続を選択します。

```
CMD$="HCSU PORT, RS, DEV, EPSON"
CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

HARDCOPY_TRANSMIT, SCREEN_DUMP

補足情報

パラメータは、ペアにグループ化されます。ペアの最初に変更する変数名を指定し、ペアの2番目に新しく割り当てる値を指定します。ペアは任意の順番に並べることも、変更する変数だけにとどめることもできます。

表　　記	
DEV	デバイス
PENS	プロッタ: プロット・ペン
PFEED	改ページ
PORT	出力ポート名
CARD	メモリ・カード
HDD	ハード・ディスク
CENT	セントロニクス・ポート
FLPY	フロッピー・ディスク
GPIB	IEEE-488 ポート
PRT	内部プリンタ
RS	RS-232-C ポート
CMDIV	内部プリンタ: cm/division
PSIZE	プロッタ: ペーパ・サイズ
デバイス名	デバイスタイプ
EPSON	Epson black & white
EPSONCOL	Epson color
HPLJ	HP Laser Jet
HPPJ	HP Paint Jet
HPTJ	HP Think Jet
HPDJ	HP Desk Jet color
HPDJBW	HP Desk Jet baw
HP7470A	HP 7470A Plotter
HP7550A	HP 7550A Plotter
TIFF	TIFF
TIFFCOMP	TIFF Compressed
TTIFFCOL	TIFF Color
BMP	BMP

ハード・コピー　HARDCOPY_TRANSMIT、HCTR コマンド

説　明

HARDCOPY_TRANSMITコマンドは、ASCIIキャラクタ文字列をそのままハード・コピー・ユニットへ送信します。これにより、ユーザは、デバイス特定制御文字シーケンスを送信してハード・コピー・ユニットを制御できます。また、スクリーン・ダンプ時にドキュメンテーション用のテキストを追加することもできます。

コマンド構文

HardCopy_TRansmit '<string>'

<string>:= 任意のASCII文字シーケンスまたはエスケープ・シーケンス

注: このコマンドは、COMM_RS232コマンドのページに説明されているエスケープ・シーケンスを受け付けられます。エスケープ・シーケンスは、ハード・コピー・ユニットへ送信される前にASCII文字コードに変換されます。

例（GPIB）

次のコードは、ドキュメンテーション・データをプリンタへ送信します。

```
CMD$="HCTR 'Data from Oct.15＼r＼n"
CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

HARDCOPY_SETUP, SCREEN_DUMP

表示

HOR_MAGNIFY、HMAG

コマンド / クエリ

説　明

HOR_MAGNIFYコマンドは、選択した拡張トレースを水平方向に指定ファクタ分拡大します。許容範囲外の拡大ファクタは、最隣接有効値に丸められます。

マルチ・ズームがイネーブルの場合は、全拡張トレースの拡大ファクタが指定ファクタに設定されます。任意の拡張（カレント・ソース依存）に対して指定ファクタが大きすぎる場合は、許容値に削られてからトレースへ適用されます。

有効範囲外のファクタが指定されると、STBレジスタのVABビット（ビット2）が設定されます（ページSC-155の表を参照してください）。

HOR_MAGNIFY?クエリは、指定拡張機能に対する現在の拡大ファクタを返します。

コマンド構文

<exp_trace>:Hor_MAGnify <factor>

<exp_trace>:={TA, TB, TC, TD}

<factor>:=1～20000

クエリ構文

<exp_source>:Hor_MAGnify?

応答フォーマット

<exp_source>:Hor_MAGnify <factor>

例（GPIB）

下の例は、トレースA（TA）を水平方向に5倍分拡大します。

```
CMD$="TA:HMAG5":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

DUAL_ZOOM, MULTI_ZOOM

説　明

HOR_POSITIONコマンドは、ソース・トレースの強調ゾーンの水平方向の中心位置を決定します。許容位置はデビジョン0～10の範囲です。ソース・トレースがシーケンス・モードで捕捉された場合は、水平シフトは、1度に1セグメントにのみ適用されます。

マルチ・ズームがイネーブルの場合は、指定トレースの指定水平位置とカレント水平位置との差異がすべての拡張トレースに適用されます。これにより水平方向位置が画面境界の左または右にはみ出すようなトレースが発生する場合は、位置の差異が調節されてからトレースに適用されます。

拡張トレースのソースがシーケンス波形であり、マルチ・ズームがイネーブルの場合は、指定トレースの指定セグメントとカレント・セグメントの差異がすべての拡張トレースに適用されます。これによりセグメントがソース・セグメント数の範囲を超えるような拡張トレースが発生する場合は、その差異が調節されてからトレースに適用されます。

有効範囲外の値が指定されると、STBレジスタのVABビット（ビット2）が設定されます（ページSC-155の表を参照してください）。

HOR_POSITION?クエリは、ソース・トレースの強調ゾーンの中心位置を返します。

注:セグメント番号0には、"未拡張セグメントをすべて表示（Show All Segments Unexpanded）"という特別な意味があります。

コマンド構文

<exp_trace>:Hor_POSition <hor_position>,

<exp_trace>:={TA, TB, TC, TD}

<hor_position>:= 0～10 DIV

:= 0～最大セグメント

注1: 接尾文字DIVはオプションです。

注2: セグメント番号は、シーケンス・モードで捕捉された波形にのみ関連し、シングル波形捕捉では無視されます。セグメント番号が0に設定されていると、すべてのセグメントが表示されます。

クエリ構文

<exp_trace>:Hor_POSition?

応答フォーマット

<exp_trace>:Hor_POSition <hor_position>[,]

注3: セグメント番号は、シーケンス波形に対してのみ割り当てられます。

例（GPIB）

下の例は、トレースA（TA: Trace A）で現在表示されているトレースの強調ゾーンの中心位置をデビジョン3にします。

```
CMD$="TA:HPOS 3":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

DUAL_ZOOM, MULTI_ZOOM

クエリ

説　明

*IDN?クエリは機器情報の問い合わせです。応答は4フィールド構成でされ、製造者、オシロスコープ・モデル、シリアル番号、ファームウェア・レビジョン・レベルに関する情報を提供します。

クエリ構文

*IDN?

応答フォーマット

*IDN LECROY,<model>,<serial_number>,<firmware_level>

<model>:=6文字または7文字のモデルID

<serial_number>:9桁または10桁の10進数コード

<firmware_level>:= リリース・レベルの2桁数字に、ピリオド、マイナー・リリース・レベル1桁、ピリオド、更新レベル1桁が続きます（xx.y.z）。

例（GPIB）

この例は、オシロスコープにID要求を発行します。

```
CMD$="*IDN?":CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
```

応答メッセージ:

*IDN LECROY, LT344L LT34401000, 8.1.0

ステータス　INE

コマンド / クエリ

説　明

INEコマンドは、内部ステート変更イネーブル（INE: Internal state change Enable）レジスタを設定します。このコマンドにより、INRレジスタの複数のイベントをSTBレジスタのINBサマリ・メッセージ・ビット（ビット0）に反映できます。INR定義イベントの概略は、ページSC-93の表を参照してください。

INE?クエリは、INEレジスタの内容を読み出します。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

INE <value>

<value>:= 0〜65535

クエリ構文

INE?

応答フォーマット

INE <value>

例（GPIB）

次のコマンドにより、スクリーン・ダンプ完了（ビット1、つまり10進数の2）および/または波形捕捉完了（ビット0、つまり10進数の1）が発生したときにINBビットが設定されます。この2つの値の和は、INEマスク2 + 1 = 3となります。

```
CMD$="INE 3":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

INR

ステータス　INR?

クエリ

説　明

INR?クエリは、内部ステート変更レジスタ（INR: INternal state change Register）の読み出しとクリアを行います。INRレジスタ（下の図参照）には、さまざまな内部動作の完了とステート遷移が記録されます。

ﾋﾞｯﾄ	ﾋﾞｯﾄ値	説　　明
15...14		0　将来に備えて予約済み。
13	8192	1　トリガー・レディー状態。
12	4096	1　Pass/Fail テストで Pass または Fail が検出された。
11	2048	1　トレース D での波形処理が終了した。
10	1024	1　トレース C での波形処理が終了した。
9	512	1　トレース B での波形処理が終了した。
8	256	1　トレース A での波形処理が終了した。
7	128	0　フロッピ・ディスクかメモリ・カード、ハードディスクが交換された。
6	64	1　オート・ストア中にフロッピ・ディスクかメモリ・カード、ハードディスクが一杯になった。
5	32	1　将来に備えて予約済み。
4	16	1　シケーンス・トリガのセグメントの捕捉がなされた。
3	8	1　データ・ブロック転送にタイムアウトが起きた。
2	4	1　ローカル状態に戻っているのが見つかった。
1	2	1　スクリーン・ダンプが終了した。
0	1	1　新たな信号が補足された。

内部ステート・レジスタ（INR）の構造

クエリ構文

INR?

応答フォーマット

INR<state>

<state>:= 0～65535

例（GPIB）

次の命令は、INRレジスタの内容を読み出します。

```
CMD$="INR?":CALL IBWRT (SCOPE%, CMD$)
```

応答メッセージ:

```
INR 1026
```

トレースCでの波形処理およびスクリーン・ダンプの両方が終了しています。

関連コマンド

ALL_STATUS, *CLS, INE

説　明

INSPECT?クエリにより、ユーザは、捕捉波形をわかりやすいフォーマットで部分的に読み出すことができます。コマンドは、テンプレートで供給されている波形のフォーマット説明に従っています（TEMPLATE?クエリで最新のコピーが得られます）。波形の論理ブロックはそれぞれ、ブロック名（たとえば、テンプレートの説明にあるようなTRIGTIME）を第1（文字列）パラメータとして引用符で囲んで指定して、調べることができます。

WAVEDESCという名前の特殊論理ブロックも詳細に調べられます。WAVEDESCブロックの変数名を第1（文字列）パラメータとして引用符で囲んで指定すると、その変数の実値のみを調べることができます。

表　　記	
BYTE	整数の生データ（8m.s.b.*で切り捨て）
FLOAT	(ボルトや単位で表される測定値の)浮動小数点数で正規化されたデータ（ゲインやオフセットに適用される）
WORD	整数の生データ（16m.s.b.*で切り捨て）
	*最上位ビット

クエリ構文

<trace>:INSPect? '<string>'[,<data_type>]

<trace>:={TA, TB, TC, TD, M1, M2, M3, M4, C1, C2, C3☝, C4☝}

<string>:= 論理ブロックの有効名、またはWAVEDESCブロック内の有効変数名（TEMPLATEコマンドと第5章を参照してください）。

<data_type>:={BYTE, WORD, FLOAT}

注:オプション・パラメータ<data_type>は、データ・アレイの読み出しのみに適用され、データの表現形式を選択します。デフォルトの<data_type>はFLOATです。

応答フォーマット

<trace>:INSPect "<string>"

<string>:=論理ブロックや変数の名前と値を複数指定する文字列。

☝使用可能対象

<Trace>:C3およびC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

1) 次のコマンドは、チャネル1で最終波形が捕捉されたタイムベース値を読み出します。

```
CMD$:="C1:INSP? 'TIMEBASE'"
CALL IBWRT（SCOPE%, CMD$）
CALL IBRD（SCOPE%, RSP$）:PRINT RSP$
応答メッセージ:
C1:INSP "TIMEBASE:500 US/DIV"
```

2) 次のコマンドは、波形記述子ブロックの内容全体を読み出します。

```
CMD$:="C1:INSP? 'WAVEDESC'"
```

関連コマンド

TEMPLATE, WAVEFORM_SETUP

説　明

INTENSITYコマンドは、グリッドまたはトレース/テキストの輝度レベルを設定します。

輝度レベルは、パーセンテージ（PCT）で表されます。100 PCTは最大輝度に対応し、0 PCTは最小輝度に対応します。

INTENSITY?クエリに対する応答は、グリッドとトレースの輝度レベルを返します。

コマンド構文

INTenSity GRID,<value>, TRACE,<value>

<value>:=0～100 PCT

注1: パラメータは、ペアにグループ化されます。ペアの最初に変更する変数名を指定し、ペアの2番目に新しく割り当てる値を指定します。ペアは任意の順番に並べることも、変更する変数だけにとどめることもできます。

注2: 接尾文字PCTはオプションです。

クエリ構文

INTenSity?

応答フォーマット

INTenSity TRACE,<value>, GRID,<value>

例（GPIB）

次の命令は、輝度のリモート・コントロールをイネーブルにし、グリッド輝度レベルを75 %に変更します。

```
CMD$="INTS GRID, 75":CALL IBWRT (SCOPE%, CMD$)
```

捕　捉　　INTERLEAVED、ILVD

コマンド / クエリ

説　明

INTERLEAVEDコマンドは、シングル・ショット・モードとランダム・インターリーブ・サンプリング（RIS: random interleaved sampling）モードの両方が使用できるタイムベース設定で、RISをイネーブルまたはディスエーブルします。仕様に関しては、Operator's Manualの付録Aを参照してください。

RISはシーケンス・モード捕捉では使用できません。

INTERLEAVED?クエリは、オシロスコープがRISモードであるかどうかを返します。

コマンド構文

InterLeaVeD <mode>

<mode>:={ON, OFF}

クエリ構文

InterLeaVeD?

応答フォーマット

InterLeaVeD <mode>

例

次のコマンドは、オシロスコープにRISモードを使用するよう命令します。

CMD$="ILVD ON":CALL IBWRT (SCOPE%, CMD$)

関連コマンド

TIME_DIV, TRIG_MODE, MEMORY_SIZE

ステータス　*IST?

クエリ

説　明

*IST?（Individual STatus:個別ステータス）クエリは、IEEE 488.1定義"ist"ローカル・メッセージのカレント・ステートを読み出します。"ist"個別ステータス・メッセージは、並列ポール動作時に送信されるステータス・ビットです。

クエリ構文

*IST?

応答フォーマット

*IST <value>

<value>:=0または1

例（GPIB）

次のコマンドは、ISTビットの内容を読み出します。

```
CMD$="*IST?":CALL IBWRT (SCOPE%, CMD$)
CALL IBRD (SCOPE%, RSP$) :RSP$
```

応答メッセージ:

*IST 0

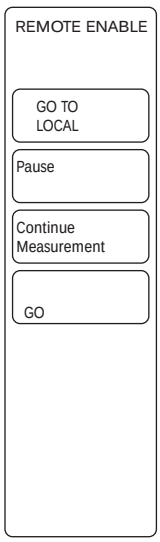
関連コマンド

*PRE

表　示　　KEY

コマンド

説　明



KEYコマンドにより、フロント・パネルからプログラムを制御できます。このフロント・パネルには、対応メニューとして1行13文字で2行までの文字列が表示され、下部の6つのメニュー・ボタンで操作できます。（1番上のメニューとボタンは自動的に"GO TO LOCAL"になります）。

オペレータがこれらのメニューに割り当てたテキスト文字列は、オシロスコープをローカル・ステートに切り替えると消えますが、リモート・ステートに戻すと再表示されます。テキストは、電源投入時、オシロスコープをリセットした場合、または空の文字列をロケーションに代入する（たとえばKEY 2,''）と、クリアされます。リモート・モード時にメニュー・ボタンを押すと、ユーザ要求ステータス・レジスタ（URR: User Request status Register）とイベント・ステータス・レジスタのURQビットが設定されます。これにより、サービス要求メカニズムがイネーブルであると、SRQが生成されます。

注:（テキストが表示されるのはリモート・モードのみですが）このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

KEY <button>,'<:string>','<string>'

<button>:=2～6

<string>:=1行13文字で2行までの文字列（任意のASCIIコード）

例（GPIB）

上の図で示したメニュー例は、以下のKEYコマンド・シリーズを発行して作成されたものです。

```
CMD$="KEY 2, 'Pause';KEY 3, 'Continue','Measurement';
KEY 4, ' ','GO'":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

URR

コマンド / クエリ

説　明

LOGO コマンドは、タイム・グリッドやXYグリッドの左上角にあるレクロイのロゴの表示、非表示の選択をします。

コマンド構文

LOGO <state>

<state>: ={ON, OFF}

クエリ構文

LOGO?

応答フォーマット

LOGO <state>

例（GPIB）

次のコマンドは、ロゴを表示させます。

```
CMD$="LOGO ON":CALL IBWRT (SCOPE%, CMD$)
```

カーソル

MASK

コマンド / クエリ

説　明

For PolyMask:

MASK COLOR コマンドで、色を2色選択できます。1色はマスク用でもう1色はマスク外のサンプルポイントを囲む円の色です。

MASK DISP_FILLED コマンドは、マスクを満たすかどうかを選択します。試験のあいだは、このコマンドの設定に関係なくすべてマスク状態になります。

MASK DRAWTOコマンドは、 現在位置から新しい位置へ線を引きます。コマンドのみでクエリはありません。

MASK ERASE コマンドは、現在のマスクを消去します。

MASK FILL コマンドは、スタートした点から囲まれたポリグラムを満たします。

MASK MOVETO コマンドは線を引くことなしにカーソルを新しい位置へ移動させます。 コマンドのみでクエリはありません。

SHOW_FAIL コマンドはマスク内外のエラーを円で囲むかどうかを設定します。

コマンド構文

<destination>:MASK COLOR <mask colors>,<error color>

カラーコマンド下にある、カラー選択表を参照してください。

<destination>:MASK DISP_FILLED <state>

<state>: ={YES,NO}

<destination>:MASK DRAWTO <x_value>,<y_value>

<x_value>: =0 ～ 10 div (XYプロット時：-4 ～ +4 div)

<y_value>: =-4～ +4 div

<destination>:MASK ERASE

<destination>:MASK FILL <x_value>,<y_value>

<x_value>: =0 ～10 div

<y_value>: =-4 ～ +4 div

<destination>:MASK MOVETO <x_value>,<y_value>

<x_value>: =0 ～ 10 div

<y_value>: =-4 ～ +4 div

<destination>:SHOW_FAIL <state>,<count>

<state>: ={OFF, INSIDE, OUTSIDE}

<count>: =1 ～ 1000

<destination>: ={C1, C2, C3, C4, TA, TB, TC, TD, TXY, PMXY}

クエリ構文

MASK? COLOR

MASK? DISP_FILLED

例（GPIB）

TA:MASK COLOR,BLUE,RED

表　示

MEASURE_GATE、MGAT☝

コマンド / クエリ

説　明

MEASURE_GATEコマンドは、パラメータ測定ゲート領域（パラメータ・カーソル間の領域）をハイライト表示するかどうか制御します。ハイライト表示は、測定ゲート領域外のトレース範囲をニュートラル・カラーにすることで行われます。

MEASURE_GATE?クエリに対する応答は、パラメータ測定ゲート領域がハイライト表示されているかどうかを返します。

コマンド構文

Measure_GATe <state>

<state>:={ON, OFF}

クエリ構文

Measure_GATe?

応答フォーマット

Measure_GATe <state>

☝使用可能対象

コマンド/クエリは、LC-Series（LCシリーズ）のオシロスコープでのみ使用できます。

例（GPIB）

次の命令は、測定ゲート領域をハイライト表示します。

```
CMD$="MGAT ON":CALL IBWRT (SCOPE%, CMD$)
```

捕　捉

MEMORY_SIZE、MSIZ☝

コマンド / クエリ

説　明

MEMORY_SIZEコマンドにより、捕捉に用いる最大メモリ長を選択できます。データポイント数を減らすと、スループットが高速になります。

MEMORY_SIZE?クエリは、波形捕捉に用いられる現在の最大メモリ長を返します。

コマンド構文

Memory_SIZe <size>

<size>:={500, 1K, 2.5K, 5K, 10K, 25K, 50K, 100K, 250K, 500K, 1M, 2.5M}

注:オシロスコープは、要求<size>を使用可能チャネル・メモリに適用します。

クエリ構文

Memory_SIZe?

応答フォーマット

Memory_SIZe <size>

☝使用可能対象

203ページのモデル別：使用可能対象一覧をご参照ください。

例

次のコマンドは、シングル・ショット捕捉あたりまたはRIS捕捉当たりのオシロスコープの捕捉データ・サンプルを最大10000に設定します。

```
CMD$="MSIZ 10K":CALL IBWRT (SCOPE%, CMD$)
```

表　示

MESSAGE、MSG

コマンド / クエリ

説　明

MESSAGEコマンドは、グリッド上部のメッセージ・フィールドに文字列を表示します。文字列は最長49文字です。文字列は、オシロスコープがリモート・モードであって、内部ステータス・メッセージが生成されない限り表示されたままです。オシロスコープをローカル・モードに戻すと、メッセージは消えます。オシロスコープを再びローカルからリモートへ切り替えると、メッセージは再表示されます。メッセージは、電源投入、オシロスコープリセット、または空の文字列（MSG""）の送信によりクリアされます。

MESSAGE?クエリにより、ユーザは、最後に送信したメッセージを読み込めます。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

MeSsaGe '<string>'

<string>:=最大49文字の文字列

クエリ構文

MeSsaGe?

応答フォーマット

MeSsaGe "<string>"

例（GPIB）

次のコードにより、メッセージ"*Connect Probe 1*"がメッセージ・フィールドに表示されます。

```
CMD$="MSG '*Connect Probe 1*'":CALL IBWRT (SCOPE%, CMD$)
```

コマンド / クエリ

説　明

MULTI_ZOOMをONに設定すると、すべての拡張トレースに対する水平方向の拡大率と位置を同時に制御できます。このコマンドは、すべての拡張トレースの内容を同時に調べるのに便利です。

MULTI_ZOOM?クエリは、マルチ・ズームがイネーブルかディスエーブルかを返します。

注: このコマンドは、DUAL_ZOOMと同じ効力を持ちます。

コマンド構文

Multi_ZOoM <mode>

<mode>:={ON, OFF}

クエリ構文

Multi_ZOoM?

応答フォーマット

Multi_ZOoM <mode>

例（GPIB）

次の例は、マルチ・ズームをオンにします。

```
CMD$="MZOM ON":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

HOR_MAGNIFY, HOR_POSITION, DUAL_ZOOM

捕　捉

OFFSET、OFST

コマンド / クエリ

説　明

OFFSETコマンドにより、指定入力チャネルの垂直オフセットを調節できます。

最大範囲は、固定感度設定により異なります。仕様は、取扱説明書を参照して下さい。

範囲外の値を入力すると、オシロスコープは最隣接許容値に設定され、STBレジスタのVABビット（ビット2）が設定されます。

注: プローブ減衰ファクタは、オフセット調節では考慮されません。

OFFSET?クエリは、指定チャネルのDCオフセット値を返します。

コマンド構文

<channel>:OFfSeT <offset>

<channel>:={C1, C2, C3☝, C4☝}

<offset>:= 仕様は、Operator's Manualの付録Aを参照してください。

注: 接尾文字Vはオプションです。

クエリ構文

<channel>:OFfSeT?

応答フォーマット

<channel>:OFfSeT <offset>

☝使用可能対象

<channel>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、チャネル2のオフセットを-3 Vに設定します。

```
CMD$="C2:OFST -3V":CALL IBWRT (SCOPE%, CMD$)
```

コマンド / クエリ

説　明

ゲイン変更時、このコマンドは、オフセットを volts または division に固定します。

コマンド構文

OFfset_ConstanT <VOLTS,DIV>

クエリ構文

OFCT?

応答フォーマット

OFCT VOLTS

ステータス　*OPC

コマンド / クエリ

説　明

*OPC（OPeration Complete: 動作完了）コマンドは、標準イベント・ステータス・レジスタ（ESR: Event Status Register）のOPTビット（ビット0）をTRUEに設定します。オシロスコープは前のコマンドやクエリの処理を完了してから次のコマンドやクエリの解析を開始するため、このコマンドによるオシロスコープ動作への影響は、上記以外にはありません。

オシロスコープは前のコマンドの実行をすべて完了してからクエリに対して応答するので、*OPC?クエリに対する応答は常にASCII文字"1"です。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

*OPC

クエリ構文

*OPC?

応答フォーマット

*OPC 1

関連コマンド

*WAI

クエリ

説　明

*OPT?クエリはオシロスコープに搭載されているオプションを特定するために使用します。このクエリに対する応答は、搭載されているすべてのオプションの一覧を返します。

クエリ構文

*OPT?

応答フォーマット

*OPT <option_1>,<option_2>,..,<option_N>

<option_n> : =3文字もしくは4文字のASCII文字列

例（GPIB）

インストールされているオプションを確認します。

```
CMD$="*OPT?": CALL IBWRT(SCOPE%,CMD$):
CALL IBRD(SCOPE%,RSP$): PRINT RSP$
```

もし、オプションとしてEMMと内蔵プリンタ(GP02)及びメモリ・カードオプション(MC01)が搭載されていた場合は、以下のような応答が得られます。

```
*EMM, GP02, MC01
```

補足情報

表　　記	
EMM	拡張演算パッケージ・オプション
WAVA	波形解析パッケージ・オプション
JTA	ジッタ・タイミング解析パッケージ・オプション
MC01	メモリ・カード(SRAM カード)オプション
HDS	ハード・ディスク/Flash ATAカードオプション
GP02	内蔵プリンタ・オプション

保存/リコール・セットアップ　PANEL_SETUP、PNSU

コマンド / クエリ

説　明

PANEL_SETUPコマンドは、*SAV/*RSTコマンドを補完します。PANEL_SETUPコマンドにより、パネル・セットアップ・データを符号化形式で外部記憶装置にアーカイブできます。

PNSU?クエリで読み出されたセットアップ・データのみオシロスコープへリコールできます。セットアップ・データ・ブロックに間違ったデータがあると、パネル・セットアップ・エラー（ページSC-75の表を参照してください）が生成されます。

注:(CFMT、CHDR、CHLP、CORD、WFSUの各コマンドにより修正された）通信パラメータと、ステータス・レポート・システム（SRE、PRE、ESE、INE）に関連するイネーブル・レジスタは、このコマンドでは保存できません。

コマンド構文

PaNel_SetUp <setup>

<setup>:=PNSU?により以前に読み出されたセットアップ・データ・ブロック

クエリ構文

PaNel_SetUp?

応答フォーマット

PaNel_SetUp <setup>

例（GPIB）

1. 次の命令は、オシロスコープのカレント・パネル・セットアップ・データをPANEL.SETファイルに保存します。

   FILE$="PANEL.SET":CMD$="PNSU?"

   CALL IBWRT（SCOPE, CMD$）:CALL IBRDF（SCOPE%, FILE$）

2. 次のコマンドは、前にPANEL.SETファイルに保存したフロント・パネル・セットアップ・データをオシロスコープにリコールします。

   CALL IBWRTF（SCOPE%, FILE$）

関連コマンド

*RCL, *SAV

説　明

PARAMETER_CLRコマンドは、カスタム（Custom）モードと合否（Pass/Fail）モードで使用される5行リストのカレント・パラメータをすべてクリアします。

注: このコマンドは引数を指定しないと、PASS_FAIL_CONDITIONコマンドと同じ効力を持ちます。

コマンド構文

PArameter_CLear

関連コマンド

PARAMETER_DELETE, PARAMETER_VALUE,
PASS_FAIL_CONDITION

カーソル　PARAMETER_CUSTOM、PACU

コマンド / クエリ

説　明

PARAMETER_CUSTOM コマンドは、自動的に測定される波形パラメータをカスタム･パラメータ1～5に定義します。また詳細な定義が必要なパラメータ（例えば、やdeltaT@levelやrise@levelなど）を制御します。

PACU コマンドで設定したパラメータは PAVA?クエリでデータを読んでください。

コマンド構文

PArameter_Custom <line>,<parameter>,<qualifier>[,<qualifier>,...]

<line> : = 1～5

<parameter> : = {PAVA?クエリでリストアップされている任意のパラメータ }

<qualificr> : =各<param>で指定されている、測定クオリファイア（詳細定義）。

<param>	定　義	<qualifier> リスト
すべてのモデルで使用可能なパラメータ		
DDLY	2信号間の遅延時間	<source1>,<source2>
PHASE	2信号間の位相差	<source1>,<edge1>,<level1>,<source2>,<edge2>,<level2>,<hysteresis>,<angular unit>
EMMオプションを搭載したモデルで追加されるパラメータ		
DC2DPOS	クロックとデータ(正パルス)間の時間差	<source1>,<clock_edge>,<level1>,<source2>,<slope2>,<level2>,<hysteresis>
DC2DNEG	クロックとデータ(負パルス)間の時間差	<source1>,<clock_edge>,<level1>,<source2>,<slope2>,<level2>,<hysteresis>
DTLEV	指定されたレベルでの 2 信号間の時間差	<source1>,<slope1>,<level1>,<source2>,<slope2>,<level2>,<hysteresis>
FLEV	指定されたレベルでの立下り時間	<source>,<high>,<low>
RLEV	指定されたレベルでの立上がり時間	<source>,<low>,<high>
TLEV	指定されたレベルまでのトリガ点からの遅延時間	<source>,<slope>,<level>,<hysteresis>
WAVAオプションを搭載したモデルで追加されるパラメータ		
FWXX	指定されたレベルでの半値幅	<source>,<threshold>
PCTL	百分単位	<source>,<threshold>
XAPK	x番目に大きいピーク値	<source>,<rank>

ここで、

<sourceN> : = {C1, C2, C3G,C4G,TA, TB, TC, TD}

<slopeN> : = {POS, NEG, FIRST}

<edgeN> : = {POS, NEG}

<clock edge> : = {POS, NEG, ALL}

<levelN> : =レベルをパーセントで指定したい場合は、1 ～ 99までの数字とPCTをつける。または、

<levelN> : = 絶対値で指定したい場合は、その数値のみ。

<hysteresis> : = 0.01～ 8 divisions

<rank> : = 1 ～ 100

<threshold> : = 0 to 100 percent

<angular unit> = {PCT, DEG, RAD}

クエリ構文

PArameter_CUstom? <line>

応答フォーマット

PArameter_Custom <line>,<parameter>,<qualifier>[,<qualifier>,...]

☝使用可能対象

<sourceN> : = {C3, C4}は4チャンネルモデルのみ使用できます。

例 1

DTLEV

コマンド例

PACU 2,DTLEV,C1,POS,345E(3,C2,NEG,(789E(3,1 DIV

クエリとその応答例

PACU? 2

PACU 2,DTLEV,C1,POS,345E-3,C2,NEG,-789E-3,1 DIV

PAVA? CUST2

C2:PAVA CUST2,789 N

例 2

DDLY

コマンド例:

PACU 2,DDLY,C1,C2

クエリとその応答例:

PACU? 2
PACU 2,DDLY,C1,C2
PAVA? CUST2
C2:PAVA CUST2,123 NS

例 3

RLEV

コマンド例

PACU 3,RLEV,C1,10 PCT,67 PCT

クエリとその応答例:

PACU? 3
PACU 3,RLEV,C1,10 PCT,67 PCT
PAVA? CUST3
C1:PAVA CUST3,23 MS

例 4

FLEV

コマンド例:

PACU 3,FLEV,C1,345E_3,122E_3

クエリとその応答例:

PACU? 3
PACU 3,FLEV,C1,345E(3,122E(3
PAVA? CUST3
C1:PAVA CUST3,23 MS

関連コマンド

PARAMETER_DELETE, PARAMETER_VALUE, PASS_FAIL_CONDITION

説　明

PARAMETER_DELETEコマンドはカスタム（Custom）モードと合否モード（Pass/Fail）で使用されるパラメータのリストから指定行でのパラメータを消去します。

表　　記		
1	Line 1	カスタムあるいは合否表示
2	Line 2	カスタムあるいは合否表示
3	Line 3	カスタムあるいは合否表示
4	Line 4	カスタムあるいは合否表示
5	Line 5	カスタムあるいは合否表示

コマンド構文

PArameter_DeLete <line>

<line>:={1,2,3,4,5}

注: このコマンドは引数を指定しないコマンド

PASS_FAIL_CONDITION<line>と同じ効力を持ちます。

例（GPIB）

下記のコマンドはリストの第三テスト条件を消去します。

```
CMD$="PADL 3":CALL IBWRT(SCOPE%, CMD$)
```

関連コマンド

PARAMETER_CLR,PARAMETER_VALUE,

PASS_FAIL_CONDITION

カーソル　　PARAMETER_STATISTICS?、PAST?

クエリ

説　明

PARAMETER_STATISTICS?クエリは、指定パルス・パラメータ・モードに対する現在の統計値と、パルス・パラメータ表示の5行すべてに対する結果タイプを返します。

表　　記	
AVG	アベレージ
CUST	カスタム・パラメータ
HIGH	最大値
HPAR	水平標準パラメータ
LOW	最小値
PARAM	各行のパラメータ定義
SIGMA	標準偏差
SWEEPS	スイープ回数
VPAR	垂直標準パラメータ
NUM_ACQ	信号捕捉回数
NUM_VALUES	測定回数

クエリ構文

PArameter_STatistics? <mode>,<result>

<mode>:={CUST,HPAR,VPAR}

<result>:={AVG,LOW,HIGH,SIGMA,SWEEPS,PARAM,NUM_ACQ,NUM_VALUES}

注: このクエリでキーワードPARAMを指定すると、<parameter_name>,<source>のペアの5つのリストが返されます。

例（GPIB）

次のクエリは、5つの標準垂直パラメータの平均値を読み出します。

```
CMD$="PAST? VPAR,AVG":CALL IBWRT(SCOPE%, CMD$):
CALL IBRD(SCOPE%, RD$):PRINT RD$
```

応答フォーマット

PAST VPAR,AVG,13V,26V,47V,1V,0V

関連コマンド

PARAMETER_VALUE

クエリ

説　明

The PARAMETER_VALUE?クエリは、指定されたトレースのパルス・パラメータ(複数)とマスク・テストのカレント値(複数)を返します。パルス・パラメータの値やマスク・テストの測定値は、トレースを表示したり、選択しなくても得ることができます。

(注)

詳細な設定(クオリファイア)が必要なパラメータに関しては、PACUコマンドで各クオリファイアの設定とCUST1～CUST5への割り付けを先に行ってください。またデータを読むためには、PAVA?クエリのパラメータとして、CUST1～CUST5を指定してください。

全てのモデルで使用可能なパラメータ					
ALL	標準電圧・時間パラメータ(*)全て	DUTY	デューティー比	PER(*)	周期
AMPL(*)	振幅	FALL(*)	立下り時間	PKPK(*)	ピーク・ピーク電圧
AREA	面積	FALL82	立下り時間(80%-20%)	RISE(*)	立上り時間
BASE	ベース	FREQ	周波数	RISE28	立上り時間(20%-80%)
CMEAN	平均値(周期性)	MAX	最大値	RMS(*)	実効値
CRMS	実効値(周期性)	MEAN(*)	平均値	SDEV(*)	標準偏差
CYCL	周期数	MIN	最小値	TOP	トップ
DLY(*)	スクリーン上最初のトランジションからトリガ点までの遅延時間	OVSN	アンダーシュート	WID(*)	パルス幅
		OVSP	オーバーシュート		
PACUコマンドで詳細な定義(クオリファイア)が必要なパラメータ					
DDLY	2信号間の遅延時間	PHASE	2信号間の位相差		

EMMオプションを搭載した場合に追加されるパラメータ					
CMEDI	中間値(周期性)	FRST	波形データ先頭時間	MEDI	中間値
CSDEV	標準偏差(周期性)	LAST	波形データ最後尾時間	PNTS	データ・ポイント数
DUR	捕捉時間				
PACUコマンドで詳細な定義(クオリファイア)が必要なパラメータ					
DC2DPOS	クロックとデータ(正パルス)間の時間差	DTLEV	指定されたレベルでの2信号間の遅延時間	RLEV	指定されたレベルでの立上がり時間
DC2DNEG	クロックとデータ(負パルス)間の時間差	FLEV	指定されたレベルでの立下がり時間	TLEV	指定されたレベルまでのトリガ点からの遅延時間

WAVAオプションを搭載した場合に追加されるパラメータ					
AVG	ヒストグラムの平均値	HMEDI	ヒストグラムの中間値	PKS	ヒストグラムのピーク数
DATA	データ	HRMS	ヒストグラムの実効値	RANGE	ヒストグラムの範囲
FWHM	ヒストグラムの半値幅	HTOP	ヒストグラムのトップ	SIGMA	ヒストグラムのシグマ
HAMPL	ヒストグラムの振幅	LOW	ヒストグラムの最低値	TOTP	データ総数
HBASE	ヒストグラムのベース	MAXP	各BINの中での最大個数		
HIGH	ヒストグラムの最高値	MODE	ヒストグラムのモード値		
PACUコマンドで詳細な定義(クオリファイア)が必要なパラメータ					
FWXX	指定されたレベルの半値幅	XAPK	x番目に大きいピーク値		
PCTL	百分位数				

パラメータ計算ステート			
AV	複数周期(最大100)に対しての平均値	OF	部分的にオーバーフロー状態
GT	実際の値は、測定値より大きい可能性あり	OK	問題無し
IV	無効なデータ(データ不足)	OU	部分的にオーバーフローかつアンダーフロー
LT	実際の値は、測定値より小さい可能性あり	PT	測定ウインドウが部分的に切り取られている
NP	パルス波形無し	UF	部分的にアンダーフロー

マスク･テスト名			
ALL_IN	波形の全ポイントがマスク内 (TRUE = 1, FALSE = 0)	SOME_IN	波形のマスク内にポイントあり (TRUE = 1, FALSE = 0)
ALL_OUT	波形の全ポイントがマスク外 (TRUE = 1, FALSE = 0)	SOME_OUT	波形のマスク外のポイントあり (TRUE = 1, FALSE = 0)

クエリ構文

<trace> : PArameter_VAlue? [<parameter>,...,<parameter>]

<trace> : = {TA, TB, TC, TD, C1, C2, C3☝,C4☝}

<parameter> : = パラメータの表を参照ください。

マスク･テストに対するクエリの代替フォーマット:

<trace> : PArameter_VAlue? <mask_test>, <mask>

<mask_test> : = {ALL_IN, SOME_IN, ALL_OUT, SOME_OUT}

<mask> : = {TA, TB, TC, TD}

応答フォーマット

<trace> : PArameter_VAlue <parameter>,<value>,

<state> [,...,<parameter>,<value>,<state>]

<value> : =単位つきの測定数値

<state> : = {OK, AV, PT, IV, NP, GT, LT, OF, UF, OU}

NOTE: <parameter> を指定しないか、またはALLを指定すると、標準電圧と標準時間パラメータの測定値を計算ステートとともに返します。

☝使用可能対象

<trace> : {C3, C4} は、4CHモデルのみ有効です。

例（GPIB）

次のクエリは、トレースB(TB)の立上がり時間を読み出します

```
CMD$="TB:PAVA? RISE": CALL IBWRT(SCOPE%,CMD$):
CALL IBRD (SCOPE%,RD$): PRINT RD$
```

応答メッセージ

```
TB:PAVA RISE,3.6E(9S,OK
```

関連コマンド

CURSOR_MEASURE, CURSOR_SET, PARAMETER_CUSTOM, PARAMETER_STATISTICS

カーソル　PASS_FAIL_COUNTER、PFCT

コマンド / クエリ

説　明

PASS_FAIL_COUNTERコマンドは、Pass/Failカウンタをリセットします。PASS_FAIL_ COUNTER?クエリは、カレント・カウントを返します。

コマンド構文

Pass_Fail_CounTer

クエリ構文

Pass_Fail_CounTer?

応答フォーマット

Pass_Fail_CounTer <pass/fail>,<value>, OF,<value>

<value>:=0～999999

<pass/fail>:={PASS, FAIL}

例（GPIB）

次のクエリは、カウンタを読み出します。

CMD$="PFCT?":CALL IBWRT (SCOPE%, CMD$)

応答メッセージ:

PFCT PASS, 8, OF, 9

関連コマンド

CURSOR_MEASURE, CURSOR_SET, PASS_FAIL_DO,

PASS_FAIL_MASK, PARAMETER_VALUE

コマンド / クエリ

説　明

PASS_FAIL_DOコマンドは、Pass/Fail試験後オシロスコープに実行させる出力と動作を定義します。PASS_FAIL_DO?クエリは、現在選択されているアクションを返します。

表　　記	
BEEP	ブザー発音
PULS	背面の BNC コネクタよりパルス出力
SCDP	ハード・コピー作成
STO	メモリまたは保存媒体に格納
STOP	捕捉停止

コマンド構文

Pass_Fail_DO [<outcome>[,<act>[,<act>...]]]

<outcome>:={PASS, FAIL}

<act>:={STOP, SCDP, STO}

注1: PFDOコマンドは引数を指定しないと（"PFDO"）、動作をすべて削除ます。

注2: STOコマンドは、User's ManualのWaveform Storeの章に説明されているように保存動作を実行します。

注3: 合否がすべて検出されると、オシロスコープのINRビット12が設定されます。

クエリ構文

Pass_Fail_DO?

応答フォーマット

Pass_Fail_DO [<pass_fail>[,<act>[,<act>...]]]

例（GPIB）

次のコマンドにより、オシロスコープは試験Pass時に捕捉を停止します。

```
CMD$="PF DO PASS, STOP":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

BUZZER, CURSOR_MEASURE, CURSOR_SET, INR,
PARAMETER_VALUE, PASS_FAIL_COUNTER,
PASS_FAIL_MASK, X REAR_OUTPUT.

カーソル　PASS_FAIL_MASK、PFMS

コマンド

説　明

PASS_FAIL_MASKコマンドは、指定トレース周囲にトレランス・マスクを生成し、そのマスク・データを指定メモリに保存します。

コマンド構文

Pass_Fail_MaSk [<trace>[,<htol>[,<vtol>[,<mask>]]]

<trace>:={TA, TB, TC, TD, M1, M2, M3, M4, C1, C2, C3☝, C4☝}

<htol>:=0.0～5.0

<vtol>:=0.0～4.0

<mask>:={M1, M2, M3, M4}

注: 引数を省略すると、その引数の前の設定が適用されます。

代替コマンド形式:

Pass_Fail_MaSk INVT [,<mask>]

上記コマンドは、指定マスク・メモリのマスクを反転します。<mask>を省略すると、M4が適用されます。

☝使用可能対象

<trace>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、チャネル1トレース周囲にトレランス・マスクを生成し、そのマスク・データをM2に保存ます。

```
CMD$="PASS_FAIL_MASK C1, 0.2, 0.3, M2":
CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

PASS_FAIL_DO, PARAMETER_VALUE

クエリ

説　明

PASS_FAIL_STATUS?クエリは、指定行番号に対するPass/Fail試験のステータスを返します。

クエリ構文

Pass_Fail_STatus? <line>

<line>:={1, 2, 3, 4, 5}

応答フォーマット

Pass_Fail_STatus <line>,<state>

<state>:={TRUE, FALSE}

例（GPIB）

次のコードは、行3に指定した合否試験条件のステートを問い合わせます。

```
CMD$="PFST? 3":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

PASS_FAIL_DO, PASS_FAIL_CONDITION, PARAMETER_VALUE

カーソル　　PASS_FAIL_CONDITION、PFCO

コマンド / クエリ

説　明

PASS_FAIL_CONDITIONコマンドは、合否試験条件またはカスタム・パラメータを、Pass/Failディスプレイまたはカスタム・パラメータ（Custom Parameter）ディスプレイの指定行に追加します。

PASS_FAIL_CONDITION?クエリは、現在の合否テスト・セットアップ・データまたはカスタム・パラメータのカレント選択を指定行に表示します。

注1: 5項目までのテスト条件（またはカスタム・パラメータ）を画面の5ディスプレイ行に指定できます。PASS_FAIL_CONDITIONコマンドは1度に1行処理します。

表　記			
GT	より大きい	LT	未満

コマンド構文

Pass_Fail_COndition [<line>,<trace>,<parameter>[,<rel_op>[,<ref_value>]]]

<line>:={1, 2, 3, 4, 5}

<trace>:={TA, TB, TC, TD, C1, C2, C3, C4}

<parameter>:= ページSC-116～124のパラメータ名表を参照してください。

<rel_op>:={GT, LT}

<ref_value>:= -1e15～+1e15

注2: PFCOコマンドは引数を指定しないと（つまり、"PFCO"）、条件をすべて削除します。PFCOコマンドに引数を1つつけると（"PFCO<line>"）、<line>の条件が削除されます。

注3: 古いマスク・テスト・キーワードALLIとANYOは、マスク波形TDに対する<trace>試験を意味します。古いマスク・テスト・キーワードINSIDEとOUTSIDEは、ALL_INとSOME_OUTと同等です。これら古いキーワードは、古いバージョンとの互換性のためにのみサポートされています。

マスク・テストに対する代替コマンド形式:

Pass_Fail_COndition [<line>,<trace>,<mask_test>,<mask>]

<mask_test>:={ALL_IN, SOME_IN, ALL_OUT, SOME_OUT}

<mask>:={TA, TB, TC, TD}

クエリ構文

PFCO? <line>

応答フォーマット

PFCP <line>,<trace>,<parameter>,<rel_op>,<ref_value>

マスク・テストに対する代替応答形式:

PFCO <line>,<trace>,<mask_test>,<mask>

☝使用可能対象

<trace>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、リストの第1テスト条件を"チャネル1の周波数10 kHz未満"に設定します。

```
CMD$="PFCO 1, C1, FREQ, LT, 10000":
CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

CURSOR_MEASURE, CURSOR_SET, PASS_FAIL_COUNTER,
PASS_FAIL_DO, PASS_FAIL_MASK, PARAMETER_VALUE

カーソル　PER_CURSOR_SET、PECS

コマンド / クエリ

説　明

PER_CURSOR_SETコマンドにより、ユーザは、6個の独立カーソルのどれでも指定画面位置に置くことができます。カーソルが現在画面に表示されていなくても、カーソル位置を変更したり照会したりできます。

PER_CURSOR_SET?クエリは、複数のカーソルの現在位置を返します。

垂直カーソル位置は、CURSOR_SETコマンドで制御する位置と同じです。

表　記			
HABS	水平絶対	VABS	垂直絶対
HDIF	水平差分	VDIF	垂直差分
HREF	水平基準	VREF	垂直基準

コマンド構文

<trace>:PEr_Cursor_Set <cursor>,<position>[,<cursor>,<position>,..., <cursor>,<position>

<trace>:={TA, TB, TC, TD, C1, C2, C3☝, C4☝}

<cursor>:={HABS, HDIF, HREF, VABS, VDIF, VREF}

<position>:=0～10 DIV（水平）、-29.5～29.5 DIV（垂直）

注1: 接尾文字DIVはオプションです。

注2: パラメータは、ペアにグループ化されます。ペアの最初に変更する変数名を指定し、ペアの2番目に新しく割り当てる値を指定します。ペアは任意の順番に並べることも、変更する変数だけにとどめることもできます。

クエリ構文

<trace>:PEr_Cursor_Set?　<cursor>[,<cursor>,...,<cursor>]

<cursor>:={HABS, HDIF, HREF, VABS, VDIF, VREF, ALL}

注3: <cursor>を指定しないと、ALLと見なされます。特殊な状況によりカーソル位置を特定できない場合、そのカーソル位置はUNDEFで返されます。

応答フォーマット

PEr_Cursor_Set <cursor>,<position>[,<curosr>,<position>,...,

<cursor>,<position>

☝使用可能対象

<trace>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコードは、HREFカーソルとHDIFカーソルをそれぞれ、チャネル2を基準として+2.6 DIVと+7.4 DIVに置きます。

```
CMD$="C2:PECS HREF, 2.6 DIV, HDIF, 7.4DIV"
```

関連コマンド

CURSOR_MEASURE, CURSOR_SET, PERSIST,
PER_CURSOR_VALUE,

カーソル　PER_CURSOR_VALUE?、PECV?

クエリ

説　明

PER_CURSOR_VALUE?クエリは、パーシスタンス・モード時、下記指定のカーソルにて測定された値を返します。

表　記			
HABS	水平絶対	VABS	垂直絶対
HREL	水平相対	VREL	垂直相対

クエリ構文

<trace>:PEr_Cursor_Value?　<cursor>[,<cursor>,...,<cursor>]

<trace>:={TA, TB, TC, TD, C1, C2, C3☝, C4☝}

<cursor>:={HABS, HREL, VABS, VREL, ALL}

注: <cursor>を指定しないと、ALLと見なされます。

応答フォーマット

<trace>:PEr_Cursor_Value <cursor>,<value>[,<cursor>,<value>,...,<cursor>,<value>]

☝使用可能対象

<trace>:C3とC−4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコードは、チャネル1の垂直相対カーソルで測定された値を返します。

```
CMD$="C1:PECV? VREL":CALL PRINT RSP$
CALL IBRD (SCOPE%, RSP$) :PRINT RSP$
```

応答メッセージ:

C1:PECV VREL, 56 MV

関連コマンド

CURSOR_MEASURE, PERSIST, PER_CURSOR_SET

説　明

PERSISTコマンドは、パーシスタンス表示モードをイネーブルまたはディスエーブルにします。

コマンド構文

PERSist <mode>

<mode>:={ON, OFF}

クエリ構文

PERSist?

応答フォーマット

PERSist <mode>

例（GPIB）

次のコードは、パーシスタンス表示をONにします。

```
CMD$="PERS ON":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

PERSIST_COLOR, PERSIST_LAST, PERSIST_SAT,
PERSIST_SETUP

表　示　PERSIST_COLOR、PECL

コマンド / クエリ

説　明

PERSIST_COLORコマンドは、パーシスタンス・トレースのカラー・レンダリング方式を制御します。

PERSIST_COLOR?クエリに対する応答は、カラー・レンダリング方式を返します。

コマンド構文

PERSist_CoLor <state>

<state>:={ANALOG, COLOR_GRADED}

クエリ構文

PERSist_CoLor?

応答フォーマット

PERSist _CoLor <state>

例（GPIB）

次の命令は、パーシスタンス・トレース・カラーを指定トレース・カラーの強度階調範囲に設定します。

```
CMD$="PECL ANALOG":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

COLOR, COLOR_SCHEME, PERSIST, PERSIST_LAST,
PERSIST_SAT, PERSIST_SETUP

コマンド / クエリ

説　明

PERSIST_LASTコマンドは、パーシスタンス・データ・マップに描かれた最終トレースを表示するかしないかを制御します。

PERSIST_LAST?クエリに対する応答は、パーシスタンス・データ・マップに描かれた最終トレースが表示されているかいないかを返します。

コマンド構文

PErsist_LasT <state>

<state>:={ON, OFF}

クエリ構文

PErsist_LasT?

応答フォーマット

PErsist_LasT <state>

例（GPIB）

次の命令は、パーシスタンス・データ・マップに描かれた最終トレースを表示します。

CMD$="PELT ON":CALL IBWRT (SCOPE%, CMD$)

関連コマンド

PERSIST, PERSIST_COLOR, PERSIST_SAT, PERSIST_SETUP

表　示　PERSIST_SAT、PESA

コマンド / クエリ

説　明

PERSIST_SATコマンドは、パーシスタンス表示のカラー・スペクトルが飽和するレベルを設定します。

レベルは、パーシスタンス・データ・マップ母集団合計のパーセンテージ（PCT）で指定します。100 PCTのレベルは、カラー・スペクトルがパーシスタンス・データ・マップの深さ全体にわたって分布することを示します。小さい値では、スペクトルは、指定パーセンテージ値で（最も明るく表示される値に）飽和します。

PERSIST_SAT?クエリに対する応答は、パーシスタンス・データ・マップの飽和レベルを返します。

コマンド構文

PErsist_SAt　<trace>,<value> [,<trace>, <value>]

<trace>:={C1, C2, C3, C4, TA, TB, TC, TD, ALL}

<value>:=0～100 PCT

注: 接尾文字PCTはオプションです。

クエリ構文

PErsist_SAt?

応答フォーマット

PErsist_SAt <trace>,<value>

例（GPIB）

次の命令は、チャネル3のパーシスタンス・データ・マップ飽和レベルを60 %に、つまり、データ・ポイントの60 %をカラー・スペクトルで表示し、残りの40 %を最も明るい色で飽和するよう設定します。

```
CMD$="PESA C3,60":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

PERSIST, PERSIST_COLOR, PERSIST_PERS, PERSIST_SETUP

説　明

PERSIST_SETUPコマンドは、パーシスタンス・モードのパーシスタンス表示期間を秒単位の選択肢より選択します。また、パーシスタンスは、すべてのトレースに対して、または画面最上部の2トレースのみに対して設定できます。

PERSIST_SETUP?クエリは、パーシスタンスのカレント・ステータスを返します。

コマンド構文

PErsist_SetUp <time>,<mode>

<time>:={0.5, 1, 2, 5, 10, 20, infinite}

<mode>:={TOP2, ALL}

クエリ構文

PErsist_SetUp?

応答フォーマット

PErsist_SetUp <time>,<mode>

例（GPIB）

次のコードは、画面最上部の2トレースに対して可変パーシスタンスを20秒にします。

```
CMD$="PESU 20, TOP2":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

PERSIST, PERSIST_COLOR, PERSIST_PERS, PERSIST_SAT

ステータス　*PRE

コマンド / クエリ

説　明

*PREコマンドは、並列ポール・イネーブル（PRE: Parallel poll Enable）レジスタを設定します。並列ポール・レジスタ（PPR: Parallel poll Register）の最下位8ビットは、STBビットを構成します。*PREコマンドにより、ユーザは、並列ポール・レジスタのどのビットを、'ist'個別ステータス・ビットに影響を与えるビットにするか指定できます。

*PRE?クエリは、PREレジスタ内容を読み出します。応答は、レジスタ・ビットのバイナリ合計に対応する10進数です。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

PPE <value>

<value>:=0～65,535

クエリ構文

*PRE?

応答フォーマット

*PRE <value>

例（GPIB）

次のコマンドにより、MAVビット（STBのビット4、つまり10進数の16）が設定されると即座に'ist'ステータス・ビットが1になります。これによりPRE値は16になります。

```
CMD$="*PRE 16":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

*IST

PROBE_CAL?、PRCA?

クエリ

説　明

PROBE_CAL?クエリは電流プローブAP015及びAP011の自動校正を実行します。校正が終わると結果を応答します。

クエリ構文

<channel>:PRobe_CAL?

応答フォーマット

PROBE_CAL <diagnostics>

<diagnostics>: = 0, 1

0 = 校正正常

☝使用可能対象

クエリは、リモートモードでのみ使用できます。

例（GPIB）

次の命令は、校正を行い結果を表示します。

```
CMD$="PROBE_CAL?":CALL IBWRT (SCOPE%, CMD$) :
CALL IBRD(SCORE%.RD$):PRINT RD$
応答メッセージ :PROBE_CAL 0
```

関連コマンド

AUTO_CALIBRATE, *CAL?

捕捉・プローブ　PROBE_DEGAUSS?、PRDG?

クエリ

説　明

PROBE_DEGAUSS? クエリは Waverunnerに接続されている電流プローブを自動消磁します。このクエリは、逆電流と交流磁界によりコアの飽和を減衰波形で消去し、この間の初期の過剰な振幅分を減らします。 消磁した後、電流プローブが接続されているチャネルのカップリング・モードがDCであればプローブ校正が行われます。

クエリ構文

<channel>:PROBE_DEGAUSS?

応答フォーマット

PROBE_DEGAUSS <diagnostics>

<diagnostics>: = 0 or 1

0 = 消磁と校正が行われました。

例（GPIB）

次のコマンドはプローブの消磁と校正を行います。

```
CMD$="PROBE_DEGAUSS?":CALL IBWRT (SCOPE%, CMD$)

CALL IBRD (SCOPE%, RMD$) ：PRINT RMD$
```

応答フォーマット (エラーがないとき)

```
PROBE_DEGAUSS   0
```

関連コマンド

PROBE_CAL?, PROBE_NAME?

プローブ　　PROBE_INFOTEXT?、PRIT?

クエリ

説　明

PROBE_INFOTEXT?クエリは、Waverunnerオシロスコープに接続されたプローブに関する情報を返します。

クエリ構文

<channel>:PROBE_INFOTEXT?

<channel>:={C1,C2,C3,C4,EX,EX10}

応答フォーマット

<channel>:PRIT"<info>"

例（GPIB）

次のコマンドは、チャネル1に接続されたプローブのテキストを取得します。

```
CMD$="C1:PROBE_INFOTEXT?":
CALL IBWRT(SCOPE%,CMD$):
CALL IBRD(SCOPE%,RD$):
PRINT RD$
```

関連コマンド

PROBE_CAL?,PROBE_DEGAUSS?

プローブ　PROBE_LIGHTS、PRBL

コマンド / クエリ

説　明

PROBE_LIGHTSコマンドは，Waverunnerに接続されているアクティブプローブHFP1000（オプション）のAutoColorID機能を使用するかしないかを制御します。

コマンド構文

PRobe_Lights <state>

<state>:={ON, OFF}

クエリ構文

PRobe_Lights?

応答フォーマット

PRobe_Lights <state>

例（GPIB）

つぎの命令は，HFP1000のAutoColorID機能を有効にします。

140CMD$="PRBL ON":CALL IBWRT (SCOPE%, CMD$)

関連コマンド

PROBE_NAME?

説　明

PROBE_NAME? クエリは、Waverunnerに接続されているプローブの名前を返します。受動プローブは、減衰率によって確認されます。

クエリ構文

<channel>:PROBE_NAME?

応答フォーマット

<channel>:PRobe_NAme"<name>"

<name> : = プローブの名前

例（GPIB）

次のコマンドは接続されているプローブの名前を確認します。

```
CMD$="PROBE_NAME?":CALL IBWRT (SCOPE%, CMD$)
CALL IBRD (SCOPE%, RD$) : PRINT RD$
```

関連コマンド

PROBE_CAL?, PROBE_DEGAUSS?

セットアップの保存/リコール　*RCL

コマンド

説　明

*RCLコマンドは、オシロスコープの全フロント・パネル・セットアップ・データをリコールし、5つある不揮発性パネル・セットアップ・データの1つを使用して、オシロスコープのステートを設定します。パネル・セットアップ0がデフォルトのパネル・セットアップに対応します。

*RCLコマンドは、*SAVコマンドと反対の効果を示します。

目的のパネル・セットアップにアクセスできない場合は、実行エラー・ステータス・レジスタ（EXR: EXecution error status Register）が設定され、標準イベント・ステータス・レジスタ（ESR: Event Status Register）のEXEビットが設定されます。

コマンド構文

*RCL <panel_setup>

<panel_setup>:=0～4

例（GPIB）

次のコードは、パネル・セットアップ3に保存されていたオシロスコープ・セットアップ・データがリコールされます。

```
CMD$="*RCL 3":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

PANEL_SETUP, *SAV, EXR

REAR_OUTPUT、ROUT

コマンド / クエリ

説　明

REAR_OUTPUT コマンドは、Waverunnerの背面パネルにあるBNCコネクタの信号出力タイプを設定するのに用います。

The REAR_OUTPUT?クエリは現在のコネクタのモードを返します。

コマンド構文

Rear_OUTput <mode>[,<level>[,<rate>]]

<mode>: ={OFF, PF, TRIG, TRDY, PULSE}

クエリ構文

Rear_OUTput?

応答フォーマット

Rear_OUTput <mode>,<level>[,<rate>]

例 (GPIB)

次のコマンドは背面BNC出力をオフにします。

CMD$="ROUT OFF":CALL IBWRT (SCOPE%, CMD$)

CALL IBWRT(SCOPE%,CMD$)

関連コマンド

PASS_FAIL_DO, CAL_OUTPUT

補足情報

OFF　: Turns off rear output

PF　: PASS/FAIL モード

PULSE : 単発信号を発生させます。

TRIG　: トリガ・アウトモード

TRDY　: 新しい波形取りこみにおいて、トリガがレディ状態であることを意味します。

波形転送　RECALL、REC コマンド

説　明

RECALLマンドは、保存装置のカレント・ディレクトリより内部メモリM1～M4のいずれかまたはすべてに、波形ファイルをリコールします。ただし、リコールできるのはBINARYフォーマットで保存された波形に限られます。

コマンド構文

<memory>:=RECall DISK,<device>,FILE,'<filename>'

<memory>:={M1, M2, M3, M4, ALL}

<device>:={CARD☝, FLPY, HDD☝}

<filename>:=8文字までの英数文字列に、ドットと3文字までの拡張子が続きます。

☝使用可能対象

<device>:CARD－メモリカードオプションを装着したオシロスコープでのみ使用できます。

<device>:HDD－ハードディスクオプションを装着したオシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、"SC1.001"という波形ファイルをメモリ・カードよりメモリM1へリコールします。

```
CMD$="M1:REC DISK,CARD,FILE,'SC1.001'":
CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

FUNCTION_STATE, SOTRE, INR

セットアップの保存/リコール　RECALL_PANEL、RCPN コマンド

説　明

RECALL_PANELマンドは、保存装置のカレント・ディレクトリよりフロント・パネル・セットアップ・データをリコールします。

コマンド構文

ReCall_PaNel DISK,<device>,FILE,'<filename>'

<device>:={CARD☝, FLPY, HDD☝}

<filename>:=8文字までの文字列に、拡張子".PNL"が続きます。

☝使用可能対象

<device>:CARD－メモリカードオプションを装着したオシロスコープでのみ使用できます。

<device>:HDD－ハードディスクオプションを装着したオシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、フロッピー・ディスクの"P012.PNL"というファイルよりフロント・パネル・セットアップ・データをリコールします。

```
CMD$="RCPN DISK, FLPY, FILE,'P012.PNL'":
CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

PANEL_SETUP, *SAV, STORE_PANEL, *RCL

セットアップの保存/リコール　*RST

コマンド

説　明

*RSTマンドは、デバイス・リセットを起動します。*RSTコマンドにより、8トレースすべてがGND線に設定され、デフォルト・セットアップがリコールされます。

コマンド構文

*RST

例（GPIB）

この例は、オシロスコープをリセットします。

```
CMD$="*RST":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

*CAL, *RCL

捕　捉

SAMPLE_CLOCK、SCLK

コマンド / クエリ

説　明

SAMPLE_CLOCKコマンドにより、ユーザは、外部タイムベースを制御できます。ユーザは、オシロスコープが外部クロックを使用して捕捉するデータポイント数を設定できます。

コマンド構文

Sample_CLocK <state>[,<recordlength>][,<coupling>]

<state>:={INT, ECL, LVO, TTL}

<recordlength>:={50, 100, 200, 500, 1K, 2K, 5K, 10K, 20K, 50K, 100K, 200K, 500K, 1M}

<coupling>:={D1M or D50}

注:レコード長は、現在使用している最大使用可能メモリ・モデルより長くは設定できません(<recordlength>パラメータの初期設定は10Kです)。

クエリ構文

Sample_CLocK?

応答フォーマット

Sample_CLocK <state>,<recordlength>

例（GPIB）

次のコマンドは、オシロスコープを1000データ・ポイント・レコードの外部クロック使用に設定します。

CMD$="SCLK ECL,1000":CALL IBWRT (SCOPE%,CMD$)

セットアップの保存/リコール　*SAV

コマンド

説　明

*SAVマンドは、オシロスコープのカレント・ステートを不揮発性内部メモリへ保存します。*SAVコマンドは、オシロスコープの全フロント・パネル・セットアップ・データを保存します。

注: このコマンドでは、通信パラメータ（COMM_FORMAT、COMM_HEADER、COMM_HELP、COMM_ORDER、WAVEFORM_SETUPの各コマンドにより変更されるパラメータ）とステータス・レコーディング・システムに関連するイネーブル・レジスタ（*SRE、*PRE、*ESE、INE）は、保存されません。

コマンド構文

*SAV <panel_setup>

<panel_setup>:=1～4

例（GPIB）

次のコマンドは、現在のオシロスコープのセットアップ・データをパネル・セットアップ3に保存します。

```
CMD$="*SAV 3":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

PANEL_SETUP, *RCL

ハード・コピー

SCREEN_DUMP、SCDP

コマンド / クエリ

説　明

SCREEN_DUMPコマンドは、オシロスコープの画面内容をハード・コピー・デバイスにダンプします。このコマンドは、オシロスコープのアクティビティをホールトします。

プリント・アウト時に印刷される日時スタンプは、コマンドが実行された時刻に対応します。

コマンド構文

SCreen_DumP

クエリ構文

SCreen_DumP?

応答フォーマット

SCreen_DumP <status>

<status>:={OFF}

例（GPIB）

次のコードは、スクリーン・ダンプを起動します。

```
CMD$="SCDP":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

INR, HARDCOPY_SETUP, HARDCOPY_TRANSMIT

表　示

SCREEN_SAVE、SCSV

コマンド / クエリ

説　明

SCREEN_SAVEコマンドは、オシロスコープの自動スクリーン・セーバを制御します。スクリーン・セーバは、あらかじめ設定された時間が経過すると、内部カラー・モニタを自動的にシャット・ダウンします。

SCREEN_SAVE?クエリに対する応答は、自動スクリーン・セーバ機能がオンかオフかを返します。

注: スクリーン・セーバが働いていても、オシロスコープは完全に機能しています。

コマンド構文

SCreen_SaVe <state>

<state>:={YES, NO}

クエリ構文

SCreen_SaVe?

応答フォーマット

SCreen_SaVe <state>

例（GPIB）

次の命令は、自動スクリーン・セーバをイネーブルにします。

```
CMD$="SCSV ON":CALL IBWRT (SCOPE%,CMD$)
```

説　明

SELECTコマンドは、表示されているトレースに対して手動表示制御の指定トレース選択します。指定トレースが表示されない場合には、環境エラー（ページSC-75を参照してください）が生成されます。

SELECT?クエリは、指定トレースの選択ステータスを返します。

コマンド構文

<trace>:SELect

<trace>:{TA, TB, TC, TD}

C1　C2　C3　C4も可

クエリ構文

<trace>:SELect?

応答フォーマット

<trace>:SELect<mode>

<mode>:={ON, OFF}

例（GPIB）

次のコマンドは、トレースB（TB）を選択します。

```
CMD$="TB:SEL":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

TRACE

捕　捉

SEQUENCE、SEQ☝

コマンド / クエリ

説　明

SEQUENCEコマンドは、シーケンス・モード捕捉の条件を設定します。

SEQUENCE?クエリに対する応答は、シーケンス・モード捕捉の条件を返します。

コマンド構文

SEQuence <mode>[,<segments>[,<max_size>]]

<mode>:={OFF, ON}

<segments>:=2～4000, 1セグメントきざみで設定可

<max_size>:={50, 100, 250, 500, 1000, 2500, 5K, 10K, 25K, 50K, 100K, 250K, 500K, 1M, 2.5M, 5M}

クエリ構文

SEQuence?

応答フォーマット

SEQuence <mode>,<segments>,<max_size>

<mode>:={ON, OFF}

☝使用可能対象

203ページのモデル別：使用可能対象一覧をご参照ください。

例（GPIB）

次のコマンドは、セグメント・カウントを43に、最大セグメント・サイズを250サンプルに設定し、シーケンス・モードをONにします。

```
CMD$="SEQ ON,43,250":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

TRIG_MODE

説　明

SLEEPコマンドは、オシロスコープを指定された時間スリープさせ、新しいコマンドの解釈を開始しないよいに命令します。

コマンド構文

SLEEP <n>

<n>:= 0。0　～ 1000。0

注:オプションの引数<t>でスリープ時間（秒）を指定します。<t>が与えられていない場合もしくは<t>=0。0の場合は無制限にスリープします。

スリープ状態はオシロスコープにDCL(Device Clear)を送信することによってキャンセルできます。

例（GPIB）

次のコマンドはオシロスコープを3.2秒間待機させます。

CMD$="SLEEP 3.2"；:CALL IBWRT（SCOPE%, CMD$）

関連コマンド

*TRG,WAIT

ステータス　*SRE

コマンド / クエリ

説　明

*SREコマンドは、サービス・リクエスト・イネーブル（SRE: Service Request Enable）レジスタを設定します。このコマンドにより、ユーザは、STBレジスタのどのサマリ・ビットでサービス・リクエストを生成させるか指定できます。使用できるサマリ・メッセージの概略についてはページSC-157の表を参照してください。

サマリ・メッセージ・ビットは、対応するビット・ロケーションに'1'を書き込むとイネーブルになります。逆に、対応する位置に'0'を書き込むと、関連イベントによるサービスリクエスト（SRQ: service request）生成が回避されます。SREレジスタをクリアすると、SRQ割り込みがディスエーブルされます。

*SRE?クエリは、バイナリに変換するとSREレジスタのビット設定を示す値を返します。ビット6（MSS）は設定できず、したがって戻り値は常に値0（ゼロ）であることに注意してください。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

*SRE <value>

<value>:= 0～255

クエリ構文

*SRE?

応答フォーマット

*SRE <value>

例（GPIB）

次のコマンドにより、STBレジスタのMAVサマリ・ビット（ビット4つまり十進数16）またはINBサマリ・ビット（ビット0つまり十進数1）、あるいはその両方のビットが設定されると、即座にSRQが生成されます。これら2値を合計するとSREマスク16 + 1 = 17が得られます。

CMD$="*SRE 17":CALL IBWRT（SCOPE%,CMD$）

ステータス　*STB?

クエリ

説　明

*STB?クエリは、IEEE 488.1定義のステータス・レジスタ（STB）とマスタ・サマリ・ステータス（MSS: Master Summary Status）の内容を読み出します。応答は、ステータス・バイト・レジスタのビット0～5と7の値とMSSサマリ・メッセージを表します。

*STB?クエリに対する応答は、ビット6の表示がRQSメッセージではなくMSSサマリ・メッセージであること以外、直列ポールの応答と同じです。ステータス・レジスタ構造の詳細は、ページSC-167の表を参照してください。

クエリ構文

*STB?

応答フォーマット

*STB <value>

<value>:=0～255

例（GPIB）

次の命令は、ステータス・バイト・レジスタを読み出します。

```
CMD$="*STB?":CALL IBWRT (SCOPE%,CMD$)
CALL IBRD (SCOPE%,RSP$) :PRINT RSP$
応答メッセージ:
*STB 0
```

関連コマンド

ALL_STATUS, *CLS, *PRE, *SRE

補足情報

ﾋﾞｯﾄ	ﾋﾞｯﾄ値	ﾋﾞｯﾄ名	説　　明	注
7	128	DIO7	0　将来の使用のために予約	
6	64	MSS/RQS MSS=1 RQS=1	 SREにマークされたSTBの少なくとも ビットが1サービスリクエストあり	(1) (2)
5	32	ESB	1　ESRイネーブル・イベント発生	(3)
4	16	MAV	1　出力待ち行列が空でない	(4)
3	8	DIO3	0　予約	
2	4	VAB	1　コマンド・データ値適用	(5)
1	2	DIO1	0　予約	
0	1	INB	1　イネーブル内部(INternal)ステート変更発生	(6)

ステータス・バイト・レジスタ（STB: Status Byte Register）

注

(1) マスタ・サマリ・ステータス（MSS: Master Summary Status）は、オシロスコープ・リクエスト・サービスを表し、サービス・リクエスト（Service Request）ステータスは、設定されるとオシロスコープによるサービス・リクエスト発生を示します。ビット・ポジション6は、ポーリング方法により以下のように異なります。
*STB?クエリ受信時: ビット6 = MSS
シリアル・ポーリング時: ビット6 = RQS

(2) 例: SRE = 10でSTB = 10ならばMSS = 1。SRE = 010でSTB = 100ならばMSS = 0。

(3) イベント・ステータス・ビット（ESB: Event Status Bit）は、標準イベント・ステータス・レジスタ（ESR: Event Status Register）が最後に読み出されてから、またはクリアされてから、1つ以上のイネーブルIEEE 488.2イベントが発生したかしないかを示します。ESBは、イネーブル・イベントがTRUE（1）になると設定されます。

(4) メッセージ使用可能（MAV: Message AVailable）ビットは、出力待ち行列が空であるかないかを示します。データ・バイトが出力待ち行列にあると、MAVサマリ・ビットはTRUE（1）に設定されます。

(5) 値適用ビット（VAB: Value Adapted Bit）は、コマンドのデータ値に最近似有効値が適用されるとTRUE（1）に設定されます。たとえば、タイムベースが2.5 µs/divに再定義されると適用値は2µs/divになり、VABビットが設定されます。

(6) 内部ステート・ビット（INB: INternal state Bit）は、なんらかのイネーブル内部ステートになると、TRUE（1）に設定されます。詳しくは、INRクエリを参照してください。

STOP

コマンド

説　明

STOPコマンドは、信号捕捉を即時停止します。トリガ・モードがAUTOまたはNORMの場合は、それ以上捕捉しないようトリガ・モードSTOPPEDに変えます。

コマンド構文

STOP

例

次のコマンドは、捕捉プロセスを停止します。

```
CMD$="STOP":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

ARM_ACQUISTION, TRIG_MODE, WAIT

波形転送　STORE、STO

コマンド

説　明

STOREコマンドは、内部メモリM1～M4へ、または保存装置のカレント・ディレクトリへ指定トレース内容を保存します。

コマンド構文

STre[<trace>,<dest>]

<trace>:{TA, TB, TC, TD, C1, C2, C3☝, C4☝, ALL_DISPLAYED}

<dest>:{M1, M2, M3, M4, CARD☝, FLPY, HDD☝}

注:STOREコマンドを引数なしで送信すると、Store Setupで現在イネーブルになっているトレースがすべて保存さます。このセットアップは、STORE_SETUPコマンドで変更できます。

☝使用可能対象

<trace>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

<device>:CARD－メモリカードオプションを装着したオシロスコープでのみ使用できます。

<device>:HDD－ハードディスクオプションを装着したオシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、トレースA（TA）の内容をメモリ1（M1）に保存します。

```
CMD$="STO TA,M1":CALL IBWRT (SCOPE%,CMD$)
```

次のコマンドは、現在表示されている波形をすべてメモリ・カードに保存します。

```
CMD$="STO ALL_DISPLAYED, CARD":
CALL IBWRT (SCOPE%,CMD$)
```

次のコマンドは、保存セットアップ（Storage Setup）（STORE_SETUPを参照してください）に現在定義されている保存動作を実行します。

```
CMD$="STO":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

STORE_SETUP, RECALL

セットアップの保存/リコール　STORE_PANEL、STPN コマンド

説　明

STORE_PANELコマンドは、コマンド発行時に、オシロスコープの全フロント・パネル・セットアップ情報を保存装置のカレント・ディレクトリのファイルへ保存します。

注:このコマンドでは、通信パラメータ（COMM_FORMAT、COMM_HEADER、COMM_HELP、COMM_ORDER、WAVEFORM_SETUPの各コマンドにより変更されるパラメータ）とステータス・レコーディング・システムに関連するイネーブル・レジスタ（*SRE、*PRE、*ESE、INE）は、保存されません。

コマンド構文

STore_PaNel DISK, <device>, FILE,'<filename>'

<device>:{CARD☝, FLPY, HDD☝}

<filename>:=8文字までの英数文字列に、拡張子".PNL"が続きます。

注:ファイル名を指定しないと（または空文字列を指定した場合は）、オシロスコープは内部規則に従ってファイル名を生成します。

☝使用可能対象

<device>:CARD－メモリカードオプションを装着したオシロスコープでのみ使用できます。

<device>:HDD－ハードディスクオプションを装着したオシロスコープでのみ使用できます。

例（GPIB）

次のコードは、現在のオシロスコープのセットアップ・データを"DIODE.PNL"というファイル名でメモリ・カードに保存します。

```
CMD$="STPN DISK, CARD, FILE,'DIODE.PNL'":
CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

PNSU, *SAV, RECALL_PANEL, *RCL

波形転送　STORE_SETUP、STST

コマンド / クエリ

説　明

STORE_SETUPコマンドは、トレース保存方法を制御します。シングル・トレースまたは表示トレースすべてを保存用にイネーブルできます。これは、自動保存に、またはSTOREコマンドとSTOコマンドに適用されます。トレースは、保存措置が満杯になるまで（FILL）、または最も古いトレースが新しいトレースに置き換えられて連続的に（WRAP）、各捕捉が終了するたびに保存装置へ自動保存されます。

STORE_SETUP?クエリは、自動保存（Autostore）動作の現在のモード、現在のトレース選択、現在の宛先を返します。

ただし、リコールできるのは、バイナリ（BINARY）フォーマットで保存されている波形のみです。

コマンド構文

STore_SeTup[<trace>,<dest>[,AUTO,<mode>][,FORMAT,<type>]

<trace>:={TA, TB, TC, TD, C1, C2, C3☝, C4☝, ALL_DISPLAYED}

<dest>:={M1, M2, M3, M4, CARD☝, FLPY, HDD☝}

<mode>:={OFF, WRAP, FILL}

<type>:{BINARY, SPREADSHEET, MATHCAD, MATLAB}

クエリ構文

STore_SeTup?

応答フォーマット

STore_SeTup <trace>,<dest>,AUTO,<mode>

☝使用可能対象

<device>:CARD－メモリカードオプションを装着したオシロスコープでのみ使用できます。

<device>:HDD－ハードディスクオプションを装着したオシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、チャネル1の保存を選択します。これは、メモリ・カードに空きスペースがなくなるまで（AUTO,FILL）、"autostore（自動保存）"をイネーブルにします。

```
CMD$="STST C1, CARD, AUTO,FILL":
CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

STORE, INR

波形転送　STORE_TEMPLATE、STTM コマンド

説　明

STORE_TEMPLATEコマンドは、保存装置へオシロスコープの波形テンプレートを保存します。ファイル名は、"LECROYvv.TPL"形式で自動生成されます。ここで"vv"は2桁のレビジョン番号です。

注: たとえば、レビジョン2.1に対しては、LECROY21.TPLというファイル名が生成されます。

波形テンプレート詳細は、第5章を参照してください。

コマンド構文

STore_TeMplate DISK, <device>

<device>:{CARD☝, FLPY, HDD☝}

☝使用可能対象

<device>:CARD－Mメモリカードオプションを装着したオシロスコープでのみ使用できます。

<device>:HDD－ハードディスクオプションを装着したオシロスコープでのみ使用できます。

例（GPIB）

次のコードは、現在の波形テンプレートを将来の参考用にメモリ・カードに保存します。

```
CMD$="STTM DISK, CARD":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

TEMPLATE

TDISP

コマンド / クエリ

説　明

TDISPコマンドは現在時刻と日付をコントロールし、オシロスコープの画面に表示します。

CURRENT： 現在時刻と日付を表示させます。

NONE ：時刻と日付を画面から消します。

TRIGGER ：現在表示されている波形の一番上のトリガ時間を選択します。

コマンド構文

TDISP <state>

<state>: ={CURRENT,NONE,TRIGGER}

クエリ構文

TDISP?

応答フォーマット

TDISP <state>

例（GPIB）

オシロスコープの画面から日付と時刻の表示を消去します。

```
CMD$="TDISP NONE":CALL IBWRT (SCOPE%, CMD$)
```

波形転送

TEMPLATE?、TMPL?

クエリ

説　明

TEMPLATE?クエリは、完全な波形データを作成するための、さまざまな論理エンティティを正規に記述したテンプレートのコピーを生成します。特にテンプレートには、波形の記述子部分に含まれる変数が詳細に既述されています。詳しくは、第5章を参照してください。

クエリ構文

TeMPLate?

応答フォーマット

TeMPLate "<template>"

<template>:=波形構造を既述した可変長文字列

関連コマンド

INSPECT

TIME_DIV、TDIV

コマンド / クエリ

説　明

TIME_DIVコマンドは、タイムベース設定を変更します。新しいタイムベース設定は、次の接尾文字で指定します。ナノ秒はNS、マイクロ秒はUS、ミリ秒はMS、秒はS、キロ秒はKSです。範囲外の値を指定すると、STBレジスタのVABビット（ビット2）が設定されます（ページSC-155の表を参照してください）。

TIME_DIV?クエリは、現在のタイムベース設定を返します。

コマンド構文

Time_DIV <value>

<value>:= 仕様は"Operator's Manual"の付録Aを参照してください。

注: 接尾文字S（Seconds: 秒）はオプションです。

クエリ構文

Time_DIV?

応答フォーマット

Time_DIV <value>

例（GPIB）

次のコマンドは、タイムベースを500μsec/divに設定します。

```
CMD$="TDIV 500US":CALL IBWRT (SCOPE%,CMD$)
```

次のコマンドは、タイムベースを2 msec/divに設定します。

```
CMD$="TDIV 0.002":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

TRIG_DELAY, TRIG_MODE

表　示

TRACE、TRA

コマンド / クエリ

説　明

TRACEコマンドは、トレース表示をイネーブルまたはディスエーブルにします。波形が表示できない場合は、環境エラーが設定されます（ページSC-71の表を参照してください）。

TRACE?クエリは、指定トレースが表示されているいかいないかを返します。

コマンド構文

<trace>:TRAce <mode>

<trace>:={C1, C2, C3☝, C4☝, TA, TB, TC, TD}

<mode>:{ON, OFF}

クエリ構文

<trace>:TRAce?

応答フォーマット

<trace>:TRAce <mode>

☝使用可能対象

<trace>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、トレースA（TA）を表示します。

```
CMD$="TA:TRA ON":CALL IBWRT (SCOPE%,CMD$)
```

コマンド / クエリ

説　明

TRACE_OPACITYコマンドは、トレース・カラーの透過性を制御します。トレースは、塗りつぶし（それぞれのトレースが重なって互いに消し合う）モードか、透過（重なり合ったトレースがすべて見える）モードに設定できます。

TRACE_OPACITY?クエリに対する応答は、トレースが塗りつぶしモードか透過モードかを返します。

コマンド構文

Trace_OPAcity <type>

<type>:={OPAQUE, TRANSPARENT}

クエリ構文

Trace_OPAcity?

応答フォーマット

Trace_OPAcity <type>

例（GPIB）

次の命令は、トレースが重なり合っていてもそれぞれを区別できるように表示します。

CMD$="TOPA TRANSPARENT":CALL IBWRT (SCOPE%,CMD$)

表　示　TRACE_ORDER、TORD

コマンド

説　明

このコマンドを使用して、トレースの表示順を指定することができます。

コマンド構文

Trace_ORDer <slot><trace>

<slot>:=<1,2,3,4,5,6,7,8>

ここで、1は一番上です。

<trace>:=<C1,C2,C3,C4,TA,TB,TC,TD,OFF>

クエリ構文

TORD?

応答フォーマット

TORD 1,C1,2,OFF,3,OFF,4,OFF,

5,OFF,6,OFF,7,OFF,8,OFF

例（GPIB）

次のコマンドは、チャネル1より上にトレースAを表示します。

```
CMD$="TORD 1,TA,2,C1"
CALL IBWRT(SCOPE%,CMD$)
```

関連コマンド

TRACE

説　明

このコマンドを使用して、ストレージ・メディアとのファイルの転送、オシロスコープとコンピュータ間のファイルの転送を行うことができます。コマンドフォーマットは、コンピュータのファイルをストレッジ・メディアに転送するために使用され、クエリ・フォーマットは、ストレッジ・メディアのファイルをコンピュータに転送するために使用されます。

コマンド構文

TRansfer_FiLe,<device>,FILE,'name.ext',#9nnnnnnnnn,<data><crc>

<device>:={CARD,FLPY,HDD}

注: HDDは、HD01オプションが存在する場合だけ使用できます。
　　CARDは、MC01オプションが存在する場合だけ使用できます。

<n...n>:=ファイル・サイズ(バイト単位)

<data>:=ファイル・データ(任意のデータ・ブロック)

<crc>:=<data>の32ビット巡回冗長検査

注: ファイルの読取り、書込みは、カレント・ディレクトリに対してだけ行われます。

クエリ構文

TRFL? DISK,<device>,FILE,'name.ext'

応答フォーマット

TRFL #9nnnnnnnnn<file content><crc>

例（GPIB）

次のコマンドは、フロッピ・ディスクからファイルFAVORITE.DSOを読み込みます。

```
CMD$=TRFL,DISK,FLPY,FILE,'FAVORITE.DSO'
CALL IBWRT(SCOPE%,CMD$)
```

捕　捉

*TRG

コマンド

説　明

*TRGコマンドは、ARMコマンドを実行します。

注: このコマンドは、488.1のグループ実行トリガ（GET: Group Execute Trigger）メッセージと等価です。

コマンド構文

*TRG

例（GPIB）

次のコマンドは、信号捕捉をイネーブルにします。

```
CMD$="*TRG":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

ARM_ACQUISITION, STOP, WAIT

捕　捉

TRIG_COUPLING、TRCP☝

コマンド / クエリ

説　明

TRIG_COUPLINGコマンドは、指定トリガ・ソースのカップリング・モードを設定します。

注1: HFDIVカップリングでは、オシロスコープによりトリガ・スロープが自動的に設定されます。TRIG_SLOPEコマンドは、HFDIVカップリング・モードでは使用しません。

注2: HFDIVは、トリガ・セットアップ・メニューにはHFと表示されます。

TRIG_COUPLING?クエリは、選択されたソースのトリガ・カップリングを返します。

コマンド構文

<trig_source>:TRig_CouPling <trig_coupling>

<trig_source>:={C1, C2, C3☝, C4☝, EX, EX10}

<trig_coupling>:={AC, DC, HFREJ, LFREJ, HFDIV}

クエリ構文

<trig_source>:TRig_CouPling?

応答フォーマット

<trig_source>:TRig_CouPling <trig_coupling>

☝使用可能対象

<trig_source>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

<trig_coupling>:HFDIVは、LT224/322では、使用できません。

例（GPIB）

次のコマンドは、チャネル2のトリガ・ソースのカップリング・モードをACに設定します。

```
CMD$="C2:TRCP AC":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

TRIG_SELECT

捕　捉　　TRIG_DELAY、TRDL

コマンド / クエリ

説　明

TRIG_DELAYコマンドは、（画面左端に表示される）第1捕捉データ・ポイントに対して、トリガを発生させる時間を設定します。

このコマンドでは、正のトリガ・ディレイを水平方向の画面全体に対するパーセンテージで表します。正のトリガ・ディレイではトリガ発生前にデータが捕捉されるので、このモードはプリトリガ捕捉と呼ばれます。負のトリガ・ディレイは、秒単位で指定する必要があります。負のトリガ・ディレイではトリガ発生後データが捕捉されるので、このモードはポスト・トリガ捕捉と呼ばれます。

範囲（-10,000 div x 時間軸）外の値と100%を指定すると、トリガ時間は最近接限界値に設定され、STBレジスタのVABビット（ビット2）が設定されます。

TRIG_DELAY?クエリに対する応答は、第1捕捉データ・ポイントのトリガ時間を返します。正の時間は水平方向の画面全体に対するパーセンテージで、負の時間は秒単位で表されます。

コマンド構文

TRig_DeLay <value>

<value>:=0.00 PCT～100.00 PCT（プリトリガ）

-20 PS～-10 MAS（ポストトリガ）

注:接尾文字はオプションです。正数の場合はPCTと見なされます。負数の場合はSと見なされます。接尾文字MASはMs（Megaseconds: メガ秒）を示し、非常に低速なタイムベースでの著しい遅延にのみ有用です。

クエリ構文

TRig_DeLay?

応答フォーマット

TRig_DeLay <value>

例（GPIB）

次のコマンドは、トリガ・ディレイを-20秒（ポストトリガ）に設定します。

```
CMD$="TRDL -20S":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

TIME_DIV

説　明

TRIG_LEVELコマンドは、指定トリガ・ソースのトリガ・レベルを調節します。範囲外の値が指定されると、最近接限界値に調節され、STBレジスタのVABビット（ビット2）が設定されます（ページSC-155の表を参照してください）。

TRIG_LEVEL?クエリは、カレント・トリガ・レベルを返します。

コマンド構文

<trig_source>:TRig_LeVel <trig_level>

<trig_source>:{C1, C2, C3☝, C4☝, EX, EX10}

<trig_level>:=仕様に関しては取扱説明書を参照してください。

注: 接尾文字Vはオプションです。

クエリ構文

<tirg_source>:TRig_LeVel?

応答フォーマット

<trig_source>:TRig_LeVel <trig_level>

☝使用可能対象

<trig_source>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコードは、チャネル2のトリガ・レベルを- 3.4 Vに調節します。

```
CMD$="C2:TRLV -3.4V":CALL IBWRT (SCOPE%,CMD$)
```

捕　捉

TRIG_MODE、TRMD

コマンド / クエリ

説　明

TRIG_MODEコマンドは、トリガ・モードを指定します。

TRIG_MODE?クエリは、カレント・トリガ・モードを返します。

コマンド構文

TRig_MoDe <mode>

<mode>:={AUTO, NORM, SINGLE, STOP}

クエリ構文

TRig_MoDe?

応答フォーマット

TRig_MoDe <mode>

例（GPIB）

次のコマンドは、通常モードを選択します。

CMD$="TRMD NORM":CALL IBWRT (SCOPE%,CMD$)

関連コマンド

ARM_ACQUISITION, STOP, TRIG_SELECT, SEQUENCE

説　明

TRIG_PATTERNコマンドは、トリガ・パターンを定義します。このコマンドは、パターン・ソース(Channel 1、Channel 2、Channel 3、およびChannel 4)の論理構成とトリガの発生条件を指定します。コンプレックス・トリガ・モードが有効にされていない場合でも、このコマンドを使用できることに注意してください。

LLOW	H	HIGH	
PR	PATTERN PRESENT	AB	PATTERN ABSENT
EN	PATTERN ENTERED	EX	PATTERN EXITED

TRIG_PATTERN?クエリは、現在のトリガ・パターンを返します。

注: PRとEN、ABとEXは等値です。

コマンド構文

TRig_PAttern [<source>,<state>,..<source>,<state>],STATE,<trigger_condition>

<source>:={C1,C2,C3,C4,EX}

<state>:={L,H}

<trigger_condition>:={AB,PR,EX,EN}

注:コマンドでソース・ステートを指定しない場合、ソースはX(=don't care)ステートに設定されます。このコマンドに対する応答では、Xステートは無視され、H(=high)かL(=low)のソース・ステートだけが送り返されます。

クエリ構文

TRig_PAttern? [<source>,<state>...<source>,<state>],STATE,<trigger_condition>

<source>:={C1,C2,C3,C4,EX}

<state>:={L,H}

応答フォーマット

TRig_PAttern [<source>,<state>,..<source>,<state>],STATE,<trigger_condition>

例（GPIB）

次のコマンドは、パターンの論理ステートをHLXH (CH1=H,CH2=L,CH3=X,CH4=H)として構成し、トリガ条件をパターン欠如(AB)として定義します。

```
CMD$="TRPA C1,H,C2,L,C4,H,STATE,AB":
CALL IBWRT(SCOPE%,CMD$)
```

関連コマンド

TRIG_COUPLING、TRIG_DELAY、TRIG_LEVEL、TRIG_LEVEL_2、TRIG_MODE、TRIG_SELECT、TRIG_SLOPE、TRIG_WINDOW

捕　捉

TRIG_SELECT、TRSE

コマンド / クエリ

説　明

TRIG_SELECTコマンドは、波形の捕捉にトリガをかける条件を選択します。トリガ・タイプによっては、追加パラメータを指定する必要があります。

追加パラメータは、ペアにグループ化されます。ペアの最初に変更する変数名を指定し、ペアの2番目に新しく割り当てる値を指定します。ペアは任意の順番に並べることも、変更する変数だけにとどめることもできます。

TRIG_SELECT?クエリは、現在のトリガ条件を返します。

トリガ表記			
DROP	ドロップアウト	PA	パターン
EDGE	エッジ	PL	大きいパルス
EV	イベント	PS	小さいパルス
GLIT	グリッチ	P2	パルス幅窓
HT	ホールド・タイプ	QL	クオリファイヤ
HV	ホールド値	SNG	単一ソース
HV2	第 2 ホールド値	SQ	ステート・クオリファイド
IL	長いインターバル	SR	ソース
INTV	インターバル	STD	標準（EDGE トリガ）
IS	短いインターバル	TEQ	クオリファイド・タイム / イベント
I2	インターバル幅窓	TI	時間
OFF	ホールドオフなしのウエイト	TL	時間内
注：HT と HV は、標準トリガには適用されません。			
TV トリガ表記			
CHAR	特性	ILAC	インターレース
FLD	フィールド	LINE	ライン
FLDC	フィールド・カウント	LPIC	ピクチャあたりのライン数

TVトリガ以外:

コマンド構文

TRig_SElect <trig_type>,SR,<source>,QL,<source>,
HT,<hold_type>,HV,<hold_value>,HV2,<hold value>
<trig_type>:={STD, SNG, SQ, TEQ, DROP, EDGE, GLIT, INTV, PA, TVP, TVN}
<source>:={C1, C2, C3☝, C4☝, LINE, EX, EX10, PA}
<hold_type>:={TI, TL, EV, PS, PL, IS, IL, P2, I2, OFF}
<hold_value>:=仕様に関しては、取扱説明書を参照してください。
注: 接尾文字S（seconds: 秒）はオプションです。

クエリ構文

TRig_SElect?

応答フォーマット

TRig_SElect <trig_type>,SR,<source>,HT,<hold_type>,HV,
<hold_value>, HV2,<hold_value>
注: HV2は、<hold_type>がP2またはI2の場合のみ返されます。

☝使用可能対象

<source>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、チャネル1のシングル・ソース・トリガをトリガ・ソースに選択します。ホールド・タイプとホールド値は、20 ns"より小さいパルス"を選択します。

```
CMD$="TRSE SNG,SR,C1,HT,PS,HV,20 NS":
CALL IBWRT (SCOPE%,CMD$)
```

TVトリガ:

コマンド構文

TRig_SElect TVP or TVN,SR,<source>,FLDC,<field_count>,FLD,<field>,
CHR,<characterristics>,LPIC,<lpic>,ILAC,<ilace>,LINE,<line>
<source>:={C1, C2, C3☝, C4☝, NE, EX, EX10}
<field_count>:={1, 2, 4, 8}
<field>:=1～field_count
<characteristics>:={NTSC, PALSEC, CUST50, CUST60}
<lpic>:=1～1500
<ilace>:{1, 2}
<line>:= 1～1500または0（任意のライン）
注: FLD値は現在のFLDC値に変換されます。LINE値は、現在のFLD値とCHAR値に変換されます。
<field_count>:=4, 8はLT364/Lのみで使用可能です。

クエリ構文

TRig_SElect?

応答フォーマット

TRig_SElect TV,SR,<source>,FLDC,<field_count>,FLD,<field>,CHAR,<characteristic>,
LINE,<line>

☝使用可能対象

<source>:C3とC4 - 4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、外部トリガ入力に印可された2フィールドPAL/SECAM TV信号の第1フィールドのライン17の同期パルス（負）でトリガがかかるようにトリガ・システムを設定します。

```
CMD$="TRSE TVN,SR,EX,FLDC,2,FLD,1,CHAR,PALSEC,
LINE,17":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

TRIG_COUPLING, TRIG_LEVEL, TRIG_MODE, TRIG_SLOPE, TRIG_WINDOW

捕　捉

TRIG_SLOPE、TRSL☝

コマンド / クエリ

説　明

TRIG_SLOPEコマンドは、指定トリガ・ソースのトリガ・スロープを設定します。トリガ・カップリングがHFDIVに設定されている時に（TRIG_COUPLINGを参照してください）非互換TRSL命令が受信されない場合には、環境エラー（ページSC-71を参照してください）が生成されます。

TRIG_SLOPE?クエリは、選択されたソースのトリガ・スロープを返します。

コマンド構文

<trig_source>:TRig_SLope <trig_slope>

<trig_source>:={C1, C2, C3☝, C4☝, LINE, EX, EX10}

<trig_slope>:={NEG, POS, WINDOW☝}

クエリ構文

<trig_source>:TRig_SLope?

応答フォーマット

<trig_source>:TRig_SLope <trig_slope>

☝使用可能対象

<trig_source>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、チャネル2のトリガ・スロープを負に設定します。

```
CMD$="C2:TRSL NEG":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

TRIG_SELECT, TRIG_WINDOW

コマンド / クエリ

説　明

TRIG_WINDOWコマンドは、カレント・エッジ・トリガ・ソースに窓振幅をボルト単位で設定します。窓は、エッジ・トリガ・レベル周囲の中心に置かれます。

TRIG_WINDOW?クエリは、現在の窓振幅を返します。

コマンド構文

TRig_WIndow <value>

<value>:=0～25 V（最大範囲）

注: 接尾文字Vはオプションです。

クエリ構文

TRig_WIndow?

応答フォーマット

TRig_WIndow <trig_level>

例

次のコマンドは、ウィンドウ・サイズを+0.5 Vに調節します。

```
CMD$="TRWI 0.5V":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

TRIG_LEVEL, TRIG_SLOPE

その他

*TST?

クエリ

説　明

*TST?クエリは内部自己試験を実施し、その応答に、自己試験でエラーを検出したかどうかを返します。自己試験には、すべてのチャネルのハードウェア試験や、タイムベースとトリガの回路試験も含まれます。

ハードウェア障害は、返された<status>番号の固有のバイナリ・コードで示されます。応答"0"は、障害が発生しなかったことを示します。

クエリ構文

*TST?

応答フォーマット

*TST <status>

<status>:=0 自己試験問題なし

☝使用可能対象

リモート・モードでのみ使用できます。

例（GPIB）

この例では、自己試験が実施されます。

```
CMD$="*TST?":CALL IBWRT (SCOPE%,CMD$)
CALL IBWRT (SCOPE%,RD$) :PRINT RD$
応答メッセージ（障害のない場合）:
*TST 0
```

関連コマンド

*CAL

ステータス

URR?

クエリ

説　明

URR?クエリは、ユーザ・リクエスト・ステータス・レジスタ（URR: User Request status Register）の内容を読み出し、クリアします。URRレジスタは、メニュー・フィールドで押されたボタンを示します。

リモート・モードでは、URRレジスタはKEYコマンドで起動されると、最後に押されたボタンを示します。ローカル・モードでは、URRレジスタはCALL HOSTボタンが押されたかどうかを示します。最後のURR?クエリ以降にメニュー・ボタンが押されていないと、値0が返えされます。ページSC-100の図に、オシロスコープのボタン割り当てが示されています。

値	説　　明
0	ボタンは押されていません。
1	トップ・メニュー・ボタンが押されました。
2	トップから2番目のメニュー・ボタンが押されました。
3	トップから3番目のメニュー・ボタンが押されました。
4	トップから4番目のメニュー・ボタンが押されました。
5	トップから5番目のメニュー・ボタンが押されました。
100	"Call Host" キー（ルート・メニューの最下部のボタン）が押されました。

ユーザリクエストステータス・レジスタ（URR: User Request status Register）の構造

クエリ構文

URR?

応答フォーマット

URR <value>

<value>:= 0～5、100

例（GPIB）

次の命令は、URRレジスタの内容を読み出します。

```
CMD$="URR?":CALL IBWRT (SCOPE%,CMD$)
CALL IBRD (SCOPE%,RSP$) :PRINT RSP$
応答メッセージ:
URR 0
```

関連コマンド

CALL_HOST, KEY, ALL_STATUS, *CLS

表　示

VERT_MAGNIFY、VMAG

コマンド / クエリ

説　明

VERT_MAGNIFYコマンドは、指定トレースを垂直方向に拡大します。このコマンドは、トレースが表示されていないくても実行されます。

最大許容拡大率は、トレースのデータに関連する有意ビット数により異なります。

VERT_MAGNIFY?クエリは、指定トレースの拡大率を返します。

コマンド構文

<trace>:Vert_MAGnify <factor>

<trace>:={TA, TB, TC, TD}

<factor>:=4.0E-3～50（最大）

クエリ構文

<trace>:Vert_MAGnify?

応答フォーマット

<trace>:Vert_MAGnify <factor>

例

次のコマンドは、トレースAの垂直振幅をオリジナル振幅の3.45倍に拡大します。

CMD$="TA:VMAG 3.45":CALL IBWRT (SCOPE%,CMD$)

関連コマンド

VERT_POSITION

VERT_POSITION、VPOS

コマンド / クエリ

説　明

VERT_POSITIONコマンドは、画面上の指定トレースの垂直方向位置を調節します。これは、捕捉時に得られたオリジナルのオフセット値には影響を及ぼしません。

VERT_POSITION?クエリは、指定トレースの現在の垂直方向位置を返します。

コマンド構文

<trace>:Vert_POSITION <display_offset>

<trace>:={TA, TB, TC, TD}

<display_offset>:=-5900～+5900 DIV

注:接尾文字DIVはオプションです。限度は、現在の拡大係数、ディスプレイのグリッド数、トレースの初期位置により異なります。

クエリ構文

<trace>:Vert_POSition?

応答フォーマット

<trace>:Vert_POSITION <display_offset>

例

次のコマンドは、トレースA（TA）を、捕捉時の位置に対し+3 DIV上方向にシフトします。

```
CMD$="TA:VPOS 3DIV":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

VERT_MAGNIFY

捕　捉

VOLT_DIV、VDIV

コマンド / クエリ

説　明

VOLT_DIVコマンドは、垂直感度をボルト/div単位で設定します。範囲外の値が指定されると、STBレジスタのVABビット（ビット2）はが設定されます（ページSC-155の表を参照してください）。

注: プローブ減衰係数は、垂直感度調節には考慮されていません。

VOLT_DIV?クエリは、指定チャネルの垂直感度を返します。

コマンド構文

<channel>:Volt_DIV <v_gain>

<channel>:={C1, C2, C3☝, C4☝}

<v_gain>:=仕様は"Operator's Manual"の付録Aを参照してください。

注: 接尾文字Vはオプションです。

クエリ構文

<channel>:Volt_DIV?

応答フォーマット

<channel>:Volt_DIV <v_gain>

☝使用可能対象

<channel>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のコマンドは、チャネル1の垂直感度を50 mV/divに設定します。

```
CMD$="C1:VDIV 50MV":CALL IBWRT (SCOPE%,CMD$)
```

ステータス

*WAI

コマンド

説　明

*WAI（継続待機: WAIt to continue）コマンドは、IEEE 488.2標準に必須ですが、オシロスコープは前のコマンド実行をすべて完了してから次のコマンド処理を開始するので、オシロスコープには影響を与えません。

注: このコマンドは、ローカル・モードでもリモート・モードでも実行できます。

コマンド構文

*WAI

関連コマンド

*OPC

捕　捉

WAIT

コマンド

説　明

WAITコマンドは、オシロスコープに対して、現在の捕捉を完了するまで新しいコマンドの構文解釈を開始しないように命令します。

コマンド構文

WAIT [<t>]

<t>:=0。0～1000。0

注: オプションの引数 <t> で新たな捕捉が起きない場合のタイムアウト時間（秒）を指定します。<t> が与えられていない場合、もしくは <t> = 0。0 の場合はタイムアウトしません。

例（GPIB）

```
send:"TRMD SINGLE"
loop {send:"ARM;WAIT;C1:PAVA? MAX"
      read response
      process response
      }
```

この例は、信号を次々に捕捉してそれらの最大振幅を求めます。ARMは新しいデータ捕捉を開始します。WAITコマンドは、新しく捕捉された波形に対して常に最大値評価が実行されるよう保証します。

"C1:PAVA?MAX"は、オシロスコープに対して、チャネル1の波形の最大データ値を評価するよう命令します。

関連コマンド

*TRG, SLEEP

波形転送

WAVEFORM、WF

コマンド / クエリ

説　明

WAVEFORMコマンドは、コントローラよりオシロスコープへ波形を転送し、WAVEFORM?クエリは、オシロスコープからコントローラへ波形を転送します。

WAVEFORMコマンドは、外部保存されていた波形をオシロスコープの内部メモリへ戻します。波形は、以下のような複数の区域で構成されます。

1. 記述子（DESC: DESCriptor）
2. ユーザ・テキスト（TEXT）
3. 時刻（TIME）記述子
4. データ（DAT1）ブロック
5. （オプションで）第2データ・ブロック（DAT2）

波形構造詳細は、第5章を参照してください。

注:"WAVEFORM? ALL"で取り込んだ完全な波形のみオシロスコープへ再格納できます。

WAVEFORM?クエリは、オシロスコープに対して、波形をコントローラへ転送するよう命令します。区域は、個別にクエリできます。"ALL"パラメータを指定すると、4つまたは5つの区域すべてが上の説明の順番に1ブロックで転送されます。

注:波形データ・フォーマットは、最後のWAVEFORM_SETUPコマンド、最後のCOMM_ORDERコマンド、最後のCOMM_FORMATコマンドで指定されたカレント設定により異なります。

コマンド構文

<memory>:WaveForm ALL <waveform_data_block>

<memory>:{M1, M2, M3, M4}

<waveform_data_block>:= 任意のデータ・ブロック（第6章を参照してください）

クエリ構文

<trace>:WaveForm? <block>

<trace>:={TA, TB, TC, TD, M1, M2, M3, M4, C1, C2, C3, C4}

<block>:{DESC, TEXT, TIME, DAT1, DAT2, ALL}

注:パラメータを指定しないと、ALLが指定されたと見なされます。

応答フォーマット

<trace>:WaveForm <block>,<waveform_data_block>

注:波形を再格納する場合は、応答ヘッダをディスエーブルにしておくと便利です。詳細は、COMM_HEADERコマンドを参照してください。

☝使用可能対象

<trace>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

1. 次のコマンドは、メモリ1よりDAT1ブロックを読み出し、ファイル"MEM1.DAT"へ保存します。パス・ヘッダ"M1:"もデータとともに保存します。

```
FILE$="MEM1.DAT"
CMD$="M1:WF? DAT1"
CALL IBWRT（SCOPE%,CMD$）
CALL IBRDF（SCOPE%,FILE$）
```

2. 次の例は、チャネル1の全内容を"CHAN1.DAT"ファイルへ保存します。パス・ヘッダ"C1:"はスキップし、後にデータをオシロスコープへリコールできるようにしています。

```
FILE$="CHAN1.DAT":RD$=SPACE$（3）
CMD$="CHDR SHORT;C1:WF?"
CALL IBWRT（SCOPE,CMD$）
CALL IBRD（SCOPE%,RD$） Skip first 3 characters "C1:"
CALL IBRDF（SCOPE%,FILE$） Save data in file "CHAN1.DAT"
```

3. 次の例は、例2で保存した波形データをメモリ1へリコールします。

```
FILE$="CHAN1.DAT"
CMD$="M1:"
CALL IBWRT（SCOPE%,CMD$）
CALL IBWRTF（SCOPE%,FILE$）
```

"M1:"コマンドは、アクティブ波形が"M1"であることを保証します。データ・ファイルはオシロスコープへ送信されると、まずヘッダ"WF"がチェックされ（文字"C1:"はファイル読み出し時スキップされます）、デフォルト宛先"M1"が想定されます。

関連コマンド

INSPECT, COMM_FORMAT, COMM_ORDER, FUNCTION_STATE, TEMPLATE, WAVEFORM_ SETUP, WAVEFORM_TEXT

説　明

WAVEFORM_SETUPコマンドは、コントローラへ転送する波形データ量を指定します。このコマンドは、以下のパラメータ設定を制御します。

表　記			
FP	第1ポイント（First Point）	NP	ポイント数（Number of Points）
SN	セグメント番号（Segment Number）	SP	間引き（Sparsing）

a. 間引き（SP: Sparsing）

間引きパラメータは、データ・ポイント間の間隔を以下のように定義します。

SP = 0　すべてのデータ・ポイントを送信
SP = 1　すべてのデータ・ポイントを送信
SP = 4　3データポイントおきに送信

b. ポイント数（NP: Number of Points）

ポイント数パラメータは、送信ポイント数を以下のように示します。

NP = 0　すべてのデータ・ポイントを送信
NP = 1　1データ・ポイントを送信
NP = 50　最大50データ・ポイントを送信
NP = 1001　最大1001データ・ポイントを送信

c. 第1ポイント（FP: First Point）

第1ポイント・パラメータは、送信する最初のデータ・ポイントのアドレスを指定します。シーケンス・モードで捕捉した波形については、これは指定セグメントの相対アドレスです。例を以下に示します。

FP = 0　第1データ・ポイントに対応
FP = 1　第2データ・ポイントに対応
FP = 5000　データ・ポイント5001に対応

d. セグメント番号（SN: Segment Number）

セグメント番号パラメータは、波形がシーケンス・モードで捕捉された場合に送信するセグメントを指定します。このパラメータは、非セグメント波形では無視されます。例を以下に示します。

SN = 0　すべてのセグメント
SN = 1　第1セグメント
SN = 23　セグメント23

WAVEFORM_SETUP?クエリは、現在使用中の転送パラメータを返します。

コマンド構文

WaveForm_SetUp SP,<sparsing>,NP,<number>,FP,<point>,SN,

注1: 電源投入直後は、すべての値が0（つまり、間引きなしに波形全体を送信）に設定されます。

注2: パラメータは、ペアにグループ化されます。ペアの最初に変更する変数名を指定し、ペアの2番目に新しく割り当てる値を指定します。ペアは任意の順番に並べることも、変更する変数だけにとどめることもできます。

クエリ構文

WaveForm_SetUp?

応答フォーマット

WaveForm_SetUp SP,<sparsing>,NP,<number>,FP,<point>,SN,

例（GPIB）

次のコマンドは、アドレス200から開始し、2データ・ポイントおき（SP = 3）に送信するよう指定します。

```
CMD$="WFSU SP,3,FP,200":CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

INSPECT, WAVEFORM, TEMPLATE

波形転送

WAVEFORM_TEXT、WFTX

コマンド / クエリ

説　明

WAVEFORM_TEXTコマンドは、波形が捕捉されたときの捕捉条件をドキュメント化します。テキスト・バッファは、160文字まで使用できます。

WAVEFORM_TEXT?クエリは、指定トレースのテキスト部を返します。

コマンド構文

<trace>:WaveForm_TeXt'<text>'

<trace>:={TA, TB, TC, TD, M1, M2, M3, M4, C1, C2, C3☝, C4☝}

<text>:ASCIIメッセージ（最長160文字）

クエリ構文

<trace>:WaveForm_TeXt?

応答フォーマット

<trace>:WaveForm_TeXt"<text>"

☝使用可能対象

<trace>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次の例は、トレースA（TA）をドキュメント化します。

```
MSG$='"Averaged pressure signal.
Experiment carried out Jan. 15, 94'"
CMD$="TA:WFTX"+MSG$:CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

INSPECT, WAVEFORM, TEMPLATE

表　示　　XY_ASSIGN?、XYAS?

クエリ

説　明

XY_ASSIGN?クエリは、現在XYディスプレイに割り当てられているトレースを返します。X軸および/またはY軸にトレースが割り当てられていない場合は、トレース名の代わりに値UNDEFが返されます。

クエリ構文

XY_ASsign?

応答フォーマット

XY_ASsign <X_source>,<Y_source>

<X_source>:={UNDEF, TA, TB, TC, TD, C1, C2, C3☝, C4V☝}

<Y_source>:{UNDEF, TA, TB, TC, TD, C1, C2, C3☝, C4☝}

☝使用可能対象

<X_source>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

<Y_source>:C3とC4－4チャネル・オシロスコープでのみ使用できます。

例（GPIB）

次のクエリは、X-軸とY-軸それぞれに割り当てられているトレースを探します。

CMD$="XYAS?:CALL IBWRT (SCOPE%,CMD$)

応答メッセージ例:

XYAS C1, C2

関連コマンド

TRACE

コマンド / クエリ

説　明

XY_CURSOR_ORIGINコマンドは、XYディスプレイの絶対カーソル測定の原点を設定します。

絶対カーソル値は、ポイント（0, 0）ボルト（OFF）またはXYグリッド中心（ON）のどちらを基準にしても測定できます。

XY_CURSOR_ORIGIN?クエリは、現在、絶対カーソル測定に割り当てられている原点を返します。

コマンド構文

XY_Cursor_Origine <mode>

<mode>:={ON, OFF}

クエリ構文

XY_Cursor_Origin?

応答フォーマット

XY_Cursor_Origin <mode>

例（GPIB）

次のコマンドは、絶対カーソル測定の原点をXYグリッド中心に設定します。

```
CMD$="XYCO ON":CALL IBWRT (SCOPE%,CMDS$)
```

関連コマンド

XY_CURSOR_VALUE

カーソル　XY_CURSOR_SET、XYCS

コマンド / クエリ

説　明

XY_CURSOR_SETコマンドにより、ユーザは、指定画面位置の6つのXY電圧カーソルのどのカーソル位置も設定できます。カーソル位置は、目的のカーソルが現在表示されていなくても、またXYディスプレイ・モードがオフでも変更したり照会したりできます。

XY_CURSOR_SET?クエリは、複数のカーソルの現在位置を返します。

CURSOR_SETコマンドは、時間カーソルの位置を決定します。

表　　記	
XABS	X軸の垂直絶対
XREF	X軸の垂直基準
XDIF	X軸の垂直差分
YABS	Y軸の垂直絶対
YREF	Y軸の垂直基準
YDIF	Y軸の垂直差分

コマンド構文

XY_Cursor_Set <cursor>,<position>[,<curosr>,<position>...<cursor>,<position>]

<cursor>:={XABS, XREF, XDIF, YABS, YREF, YDIF}

<position>:=-4～4DIV

注1: 接尾文字DIVはオプションです。

注2: パラメータは、ペアにグループ化されます。ペアの最初に変更する変数名を指定し、ペアの2番目に新しく割り当てる値を指定します。ペアは任意の順番に並べることも、変更する変数だけにとどめることもできます。

クエリ構文

XY_Cursor_Set? [<cursor,...<cursor>]

<cursor>:={XABS, XREF, XDIF, YABS, YREF, YDIF, ALL}

注: <cursor>を指定しないと、ALLが指定されたと見なされます。

応答フォーマット

XY_Cursor_Set <cursor>,<position>[,<cursor>,<position...,<cursor>,<position>]

例（GPIB）

次のコマンドは、XREFとYDIFをそれぞれ+3 DIVと-2 DIVに置きます。

```
CMDS$="XYCS XREF, 3DIV, YDIF, -2DIV":
CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

XY_CURSOR_VALUE, CURSOR_MEASURE, CURSOR_SET

カーソル　　XY_CURSOR_VALUE?、XYCV?

クエリ

説　明

XY_CURSOR_VALUE?クエリは、Xカーソル対Yカーソルのカレント値を返します。X対Yトレースを表示しなくても、それぞれのパラメータは得られますが、有効ソースはX軸とY軸に割り当てる必要があります。

表　　記	
<cursor type>:=[HABS, HREL, VABS, VREL}	
<cusor type>_X	X
<cursor type>_Y	Y
<cursor type>_RATO	ΔY/ΔX
<cursor type>_PROD	ΔY*ΔX
<cursor type>_ANGLE	arc tan(ΔY/ΔX)
<cursor type>_RADIUS	sqrt(ΔX*ΔX+ΔY*ΔY)

クエリ構文

XY_Cursor_Value? [<parameter>,...<parameter>]

<parameter>:={HABS_X, HABS_Y, HABS_RATIO, HABS_PROD,
HABS_ANGLE, HABS_RADIUS, HREL_X, HREL_Y,
HREL_RATIO, HREL_PROD, HREL_ANGLE, HREL_RADIUS,
VABS_X, VABS_Y, VABS_RATIO, VABS_PROD, VABS_ANGLE,
VABS_RADIUS, VREL_X, VREL_Y, VREL_RATIO, VREL_PROD,
VREL_ANGLE, VREL_RADIUS, ALL}

注:<parameter>を指定しないと、またはALLを指定すると、測定カーソル値がすべて返されます。現在の環境でカーソル値を読み取れない場合は、値UNDEFが返されます。X軸またはY軸にトレースが割り当てられていない場合は、環境エラーが生成されます。

応答フォーマット

XY_Cursor_Value <parameter>,<value>[,...<parameter>,<value>]

<value>:=10進数値またはUNDEF

例（GPIB）

次のクエリは、絶対水平カーソルの割合、相対水平カーソルの角度、絶対垂直カーソルの積を読み取ります。

```
CMDS$="XYCV?  HABS_RATIO,HREL_ANGLE,VABS_PROD:
CALL IBWRT (SCOPE%,CMD$)
```

関連コマンド

CURSOR_MEASURE, CURSOR_VALUE, XY_CURSOR_ORIGIN

表　示　　XY_DISPLAY、XYDS

コマンド / クエリ

説　明

XY_DISPLAYコマンドは、XYディスプレイ・モードをイネーブルまたはディスエーブルにします。

XY_DISPLAY?クエリは、XYディスプレイのカレント・モードを返します。

コマンド構文

XY_DiSplay <mode>

<mode>:={ON, OFF}

クエリ構文

XY_DiSplay?

応答フォーマット

XY_DiSplay <mode>

例（GPIB）

次のコマンドは、XYディスプレイをオンにします。

```
CMD$="XYDS ON": CALL IBWRT(SCOPE%, CMD$)
```

表　示

XY_RENDER、XYRD

コマンド / クエリ

説　明

XY_RENDERコマンドは、画面上のXYプロット表示を調節します。スムーズモードでは、点はXYのペアがつながって表示されます。シャープモードでは、点をつなげることなく点として表示されます。

コマンド構文

XY_RenDer <state>

<state>: ={SHARP, SMOOTH}

クエリ構文

XY_RENDER?

応答フォーマット

XY_RENDER <state>

例（GPIB）

次のコマンドは表示をシャープモードにします。

CMD$="XY_RENDER SHARP":CALL IBWRT (SCOPE%, CMD$)

表　示　　XY_SATURATION、XYSA

コマンド / クエリ

説　明

XY_SATURATIONコマンドは、パーシスタンス表示のカラースペクトラムがXYディスプレイモードにおいてどのレベルで飽和されるかを設定します。

そのレベルは全データマップを100%としたときの指定パーセント(PCT) に換算して設定されます。レベル100%は、カラースペクトラムが パーシスタンス・データ・マップの全域にわたって広がっていることを意味します。より低い値においては、指定されたパーセンテージでスペクトラムは飽和（表示が一番明るくなる値）します。　PCT はオプションです。

XY_SAT? クエリに対する応答フォーマット XY パーシスタンス・データマップの飽和レベルを示します。

コマンド構文

XY_SAturation <value>

<value>:= 0 ～ 100 パーセント

クエリ構文

XY_SAturation?

応答フォーマット

XY_SAturation <value>

例（GPIB）

次のコマンドは、XYパーシスタンスデータマップの飽和レベルを60%に設定します。60 % of データポイントの60%はカラースペクトラムで 表示されます。, 残りの40%はもっとも明るい色で飽和されます。

```
CMD$="XYSA 60":CALL IBWRT (SCOPE%, CMD$)
```

関連コマンド

PERSIST_SAT

使用可能対象モデルとコマンド/クエリ

* 岩通モデル名 DS-4264 はレクロイモデル名 LT264 に相当します。(DS-を LT にして、DS-4xxx の頭の 4 を取る)

1

参照ページ	コマンド/クエリ	
SC-20	捕　捉	BANDWIDTH_LIMIT、BWL

使用可能対象

モデル LT224 では 200 MHz は設定できません。

2

参照ページ	コマンド/クエリ	
SC-33	捕　捉	COMBINE_CHANNELS、COMB

使用可能対象

LT364，LT364L，LT372，LT372L，LT374，LT374M，LT374L にて有効

3

参照ページ	コマンド/クエリ	
SC-105	捕　捉	MEMORY_SIZE、MSIZ

使用可能対象

LT224，322，262，264 の<size>は、100K まで

LT344，342，LT354 の<size>は、250K まで

LT344L，342L，264M，LT354M の<size>は、1M まで

LT364，372，374 の<size>は、500K まで

LT262ML，LT264ML の<size>は、2M まで

LT364L，374M，LT354L の<size>は、2.5M まで

LT372L，LT374L の<size>は、10M まで

ただし 2.5M，10M を設定しても実際に用いられる最大メモリ長は 2M，8M となります。

4

参照ページ	コマンド/クエリ	
SC-152	捕　捉	SEQUENCE、SEQ

使用可能対象

<max_size>

パラメータの最大値は、モデルによってことなります。

LT224，322 は 400 セグメント 50K まで LT342/344，LT354 は 1000 セグメント 100K まで

LT342L/344L/264M/354M は 4000 セグメント 500K まで LT364/372/374 は 1000 セグメント 250K まで
LT262ML/LT264ML/354L/364L/374M は 4000 セグメント 1M までです。
LT372L/374L は 4000 セグメント 4M までです。

機能一覧表

モデル名	チャネル数	周波数帯域幅 MHz	最高サンプリング MS/s	最大メモリ長 ポイント	チャネルコンバイン	セグメント数	セグメント時のメモリ長 ポイント
LT224	4	200	200	100k	×	400	50k
LT322	2	500	200	100k	×	400	50k
LT342	2	500	500	250k	×	1000	100k
LT342L	2	500	500	1M	×	4000	500k
LT344	4	500	500	250k	×	1000	100k
LT344L	4	500	500	1M	×	4000	500k
LT364	4	500	1000	500k	○	1000	250k
LT364L	4	500	1000	2M	○	4000	1M
LT262	2	350	1000	100k	×	400	50k
LT262ML	2	350	1000	2M	×	4000	1M
LT264	4	350	1000	100k	×	400	50k
LT264M	4	350	1000	1M	×	4000	500k
LT264ML	4	350	1000	2M	×	4000	1M
LT354	4	500	1000	250k	×	1000	100k
LT354M	4	500	1000	1M	×	4000	500k
LT354ML	4	500	1000	2M	×	4000	1M
LT372	2	500	4000	500k	○	1000	250k
LT372L	2	500	4000	8M	○	4000	4M
LT374	4	500	4000	500k	○	1000	250k
LT374M	4	500	4000	2M	○	4000	1M
LT374L	4	500	4000	8M	○	4000	4M

はじめに

10Base-T イーサネット・オプション付きの IWATSU-LeCroy Waverunner を使用すると、PC とネットワーク経由またはダイレクトに接続してオシロスコープを制御することができます。この接続はオシロスコープの後部にある T10Base-T ポートを使用します。

本章では、イーサネット・インタフェース（オプション）によってWaverunnerオシロスコープを制御する基本機能を紹介します。

GPIB コマンドの完全な仕様は「リモート・コントロール・マニアル」に掲載されています。この仕様は、GPIBインタフェースだけでなく、イーサネット・オプションによるオシロスコープの制御にも適用されます。

リモート・コマンドの規格

上記のリモート・コマンドは、IEEE 488.2* 標準に準拠しています。主として、電気的および機械部分を対象としていた IEEE 488.1 標準の拡張版と見ることができます。

接続

オシロスコープは、TCP/IP ネットワーク・インタフェースを使用してイーサネット・オプション経由で PC に接続することができます。

このクロス・ネットワーク・ケーブルを使用した接続によって、ネットワーク経由、またはオシロスコープと PC を一対一で直接接続することができます。またシステム対応のネットワーク・プリンタに印刷することができます。

ScopeExplorer

ScopeExplorer は任意の PC で使用できる接続ツールで、LeCroy デジタル・オシロスコープと Windows 95/98 または Windows NT/2000 デスクトップのインタフェース機能を提供します。ScopeExplorer は CDROM として、または http://www.lecroy.com にてインターネットから入手できます。

ScopeExplorer は、以下の接続機能を提供します。

リモート制御ターミナル

テレタイプ・ターミナルとしての働きにより、リモート制御コマンドをキーボードから入力することでオシロスコープに送信し、オシロスコープのレスポンスを表示することができます。ファイルからオシロスコープへコマン

*ANSI/IEEE 標準 488.2-1987 年（IEEE 標準コード、フォーマット、プロトコル、および共通コマンド、IEEE : The Institute of Electrical and Electronics Engineers Inc., 345 East 47th Street, New York, NY 10017, USA）

ド・シーケンスを送れます、またオシロスコープ・レスポンスを他のファイルへ送れます。

表示画面の取り込みと保管

この機能により、オシロスコープの表示画面をPCに転送し、表示/印刷することができます。一回ボタンを押すだけで、この画面をWindowsクリップボードに転送し、それ以降、任意のWindowsアプリケーションに貼り付けることができます。取り込んだ画面の周期的な自動リフレッシュをサポートしています。

パネル・ファイル（scope setup）の取り込み、保管、および再生

この機能により、オシロスコープのフロント・パネル設定全体を取得し、「ファイル名を付けて」PCに保存することができます。取得したパネル・ファイルを、後でオシロスコープに送り返して、オシロスコープ設定を再生することができます。

トレースの取り込み、保管、再生、およびASCIIへの変換

オシロスコープによって取得された波形は、PCに転送し、コンパクトなLeCroyバイナリ形式、または、Microsoft Excel、Mathsoft MathCadなどのPCベース分析ソフトで容易に使用できるASCII形式で保管することができます。取得済みのLeCroyバイナリ形式の波形は、個々にまたはデイレクトリー内の全てのファイルを自動的に、ASCIIファイルに変換することができます。

仮想フロント・パネル

オシロスコープは、Windows画面に表示される仮想フロント・パネルのすべてのつまみ、ボタンを使用して遠隔操作できます。

ActiveDSO

ActiveX標準をサポートする多様なWindowsアプリケーションによって**ActiveX™**コントロールはLeCroyオシロスコープを制御し、オシロスコープ-アプリケーション間で**データ交換**を行うことができます。MS Officeプログラム、Internet Explorer、Visual Basic、Visual C++、Visual Java、およびMatlab（v5.3）は、ActiveXコントロールをサポートする多くのアプリケーションで使用可能です。ActiveDSOはCDROMとして、またはwww.lecroy.comにて、インターネットから入手できます。

ActiveDSOによって、標準GPIBコマンドを使用するテスト・プログラムを作成することができます。

オシロスコープ・データとWindowsアプリケーションの統合を行う上で、

ActiveDSO は以下のような事が行えます。

・ オシロスコープ・データを Excel または Word にインポートすることによるレポートの生成
・ 進行中の測定結果の Microsoft Access Database への保存
・ Visual Basic、Java、C++、Excel（VBA）によるテストの自動化
・ GPIB、RS-232、またはイーサネット 10Base-T の使用

ActiveDSO コントロールはプログラミングの複雑さに煩わされることなく、制御元アプリケーションに単純で一貫したインタフェースを提供します。Excel マクロ内の 10 行に満たない VBA（Visual Basic for Applications）で、事前スケールされたリモート装置の波形データを復元することができます。

さらに、ActiveDSO コントロールは任意の OLE Automation 互換クライアントに目に見える形で埋め込んで、プログラミングなしに手操作で使用することができます。これは、Windows 95、Windows 98、または Windows NT を搭載した任意の PC 上で動作します。

コントロールを使用するには基本的な二つの方法があります。

1. OLE Automation 互換クライアントに埋め込まれた可視オブジェクト（取り込んだ表示画面を示す）としての使用。詳細は、第 3 章の「例 : Word」を参照してください。
2. スクリプト言語によってアクセスする、装置をリモート制御する不可視オブジェクトとしての使用。詳細は、第 3 章の「例 : VBA」を参照してください。

VBA（Visual Basic for Applications）は、最近の多くの Windows アプリケーションに組み込まれているプログラミング言語です。VBA は Visual Basic のサブセットで、OLE Automation サーバおよび ActiveX コントロールの使用を非常に単純なものにします。

Waverunner オシロスコープへの接続とリモート・コマンドの送信が容易であることが、以下の VBA サブルーチンでおわかりになるでしょう。

<原文例>

```
Sub LeCroyDSOTest()
  Dim o As Object
    Set o = CreateObject("LeCroy.ActiveDSOCtrl.1")
  Call o.AboutBox        'Present the control's About box
  Call o.MakeConnection("IP: 172.25.1.2) 'Connect to device on LAN
  Call o.WriteString("BUZZ BEEP", True) 'Make the DSO beep
End Sub
```

構文例 ：

Boolean controlName.WriteString

WriteStringメソッドには、以下の引数があります。

引数	**説明**
controlname	ActiveDSOコントロール オブジェクトの名前
textString	String、デバイスに送信するテキスト文字列
EOI	Boolean、戻り値がTRUEの場合EOI付きで終了

戻り値 ： 正常終了の場合TRUE、異常終了の場合FALSE

注 ：　　このメソッドは、stringコマンドを装置に送信します。

EOIがTRUEに設定された場合、デバイスはすぐにコマンドの解釈を始めます。この動作が、正常の望ましい動作です。

EOIがFALSEに設定された場合、コマンドを複数の部分に分けて送信することができます。デバイスはその最後の部分（EOIにTRUEが設定されます）を受け取ったときだけコマンドの解釈を始めます。

ホストへのWaverunnerの接続

本章では、ホスト PC または標準 10Base-T イーサネット・ネットワークへの Waverunner オシロスコープの接続について説明します。Windows NT と Windows 95 の両方のオペレーティング・システムがサポートされています。

PC 要件

Waverunner オシロスコープを遠隔操作するための PC の最低条件は以下の通りです。

- Pentium クラスの PC
- 32MB RAM
- 10MB の空きディスク容量
- Windows 95 または Windows NT
- 標準 RJ45 コネクタ付きの 10Base-T サポート・イーサネット・アダプタ

イーサネット接続

Waverunner オシロスコープは、標準の 10Base-T イーサネット接続が行われた状態で動作します。この装置はネットワークにプラグ・インするか、ホスト・コンピュータに直接に接続して操作することができます。これらの各接続には、異なるタイプのケーブルが必要です。PC にオシロスコープを直接に接続する場合はクロス・ケーブルが必要で、ネットワーク接続の場合はストレート・ケーブルを使用します。

Warunnerオシロスコープには工場で1つのIPアドレスが割り当てられています。DHCP（Dynamic Host Configuration Protocol）や他の自動アドレス解決機構はサポートされていません。接続を確立した後、ユーザはこのIPアドレスを変更することができます。

工場割当 IP アドレスは、172.25.1.2 です。

工場割当マスクは、255.255.0.0 です。

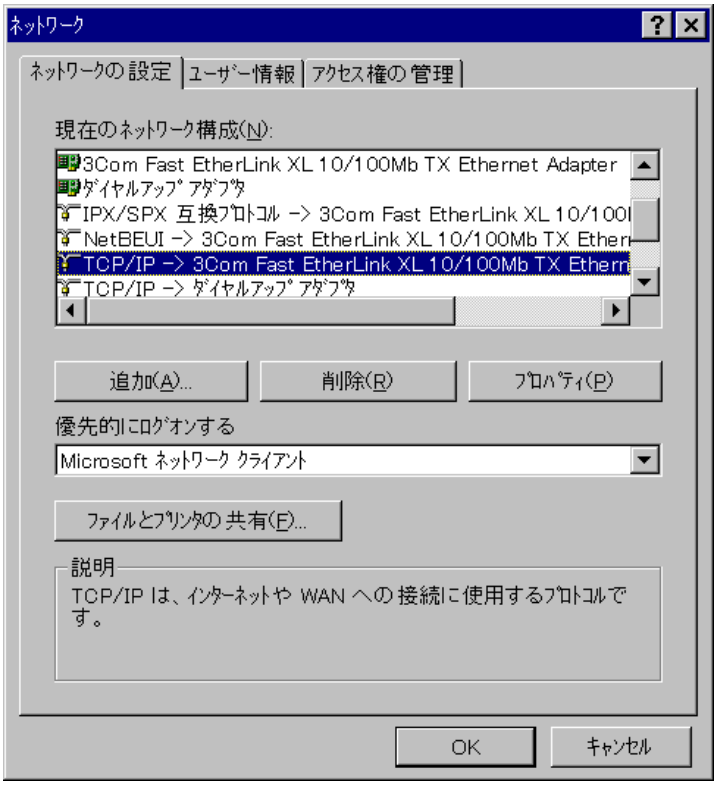
工場割当ゲートウェイは、172.25.0.1 です。

PC の設定

注 ： 次の例は、ホスト PC で Windows™ 95 が実行されていることを前提にしています。Windows NT の接続手順もこれと類似しています。

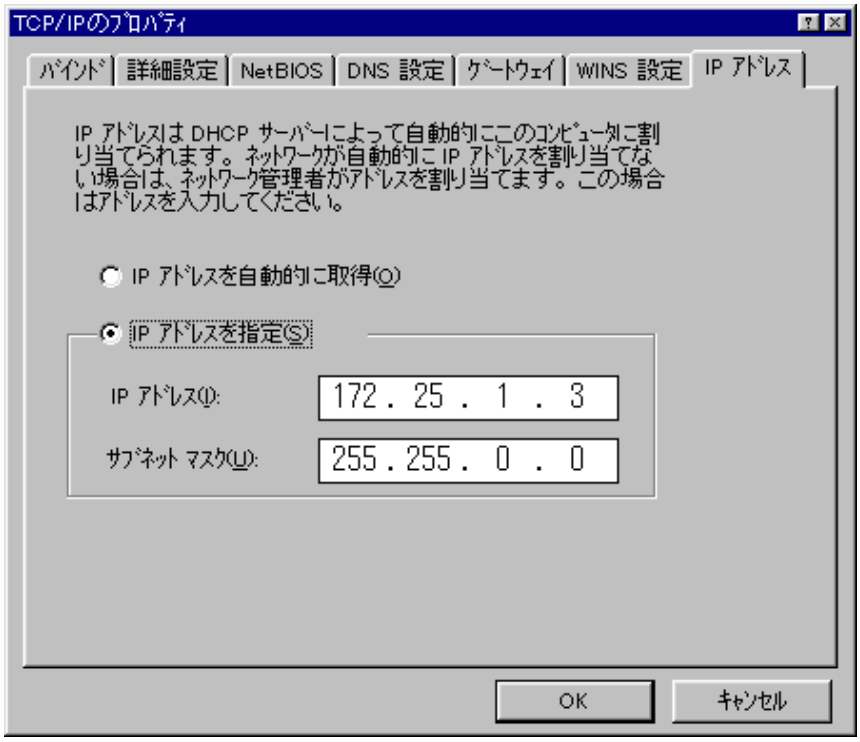
オシロスコープとホスト・コンピュータの直接接続を確立する場合、最初に、PC を正しく設定する必要があります。PC に特定の TCP/IP アドレスを割り当てます（この割当処理は、"スタティック・アドレッシング" と呼ばれます）。DHCP サーバから IP アドレスを得るように PC を設定することはできません。Windows 95 でホスト PC のスタティック・アドレスを設定するには、以下の手順に従います。

1. **[スタート] > [設定] > [コントロールパネル] を選択します。**
2. **[コントロールパネル] で、[ネットワーク] アイコンをダブルクリックして、次に示すネットワーク・ダイアログ・ボックスを表示します。**



3. **TCP/IP -> 行をダブルクリックして、次に示すダイアログ・ボックスを表示し、**

を選択します。



4. IP アドレスを指定(S) **が選択済みの場合は、コンピュータのスタティック・アドレスが設定済みで、それ以上行うことはありません。TCP/IPダイアログ・ボックスとネットワーク・ダイアログ・ボックスをキャンセルして、コントロール・パネルを閉じます。**
5. **アドレスをまだ選択していない場合は、上に示すように、IPアドレスとサブネット・マスクを入力します。IP アドレスを 172.25.x.x にした場合のサブネット・マスクは、255.255.0.0 です。コンピュータをネットワークにプラグ・インしない場合は、上記のアドレス（または、選択したサブネット内のほとんどすべてのアドレス）でかまいません。制御対象のオシロスコープのアドレスと同一のアドレスは指定できません。**
6. **次に、[TCP/IP のプロパティ] ダイアログ・ボックスで** OK **をクリックします。オペレーティング・システムおよびそのバージョンによっては、コンピュータの再起動が必要になることがあり、その場合は、ダイアログ・ボックスがそのことについて警告を出します。**

物理的接続

オシロスコープとホスト・コンピュータ間で物理的接続を確認するには、以下の手順に従います。

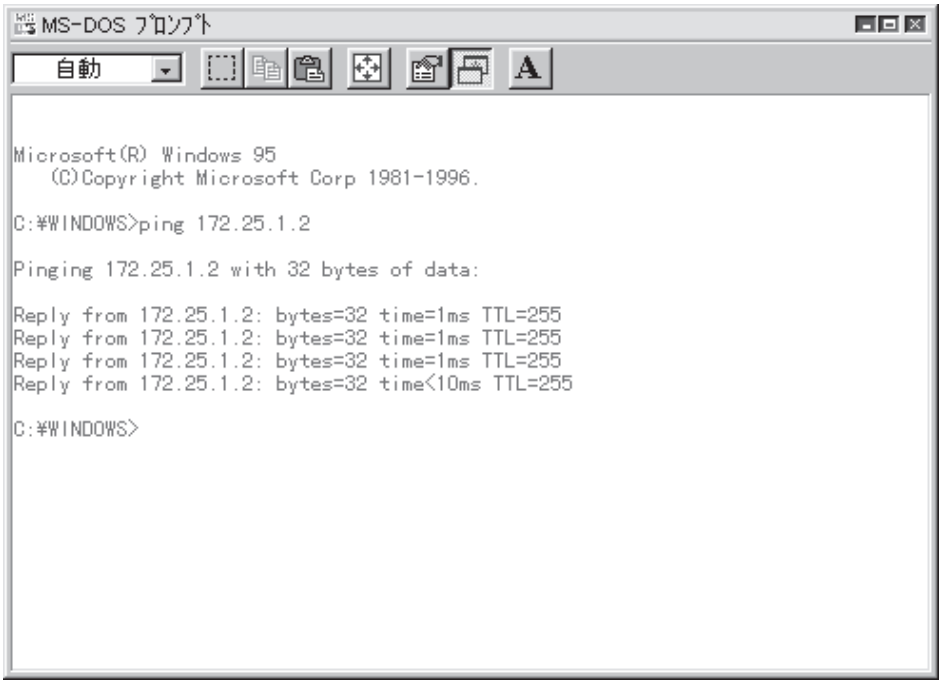
1. **クロス・ケーブルを使用して、オシロスコープを PC に接続します（直接接続の場合）。**
2. **オシロスコープの電源をオンにします。**

接続の確認

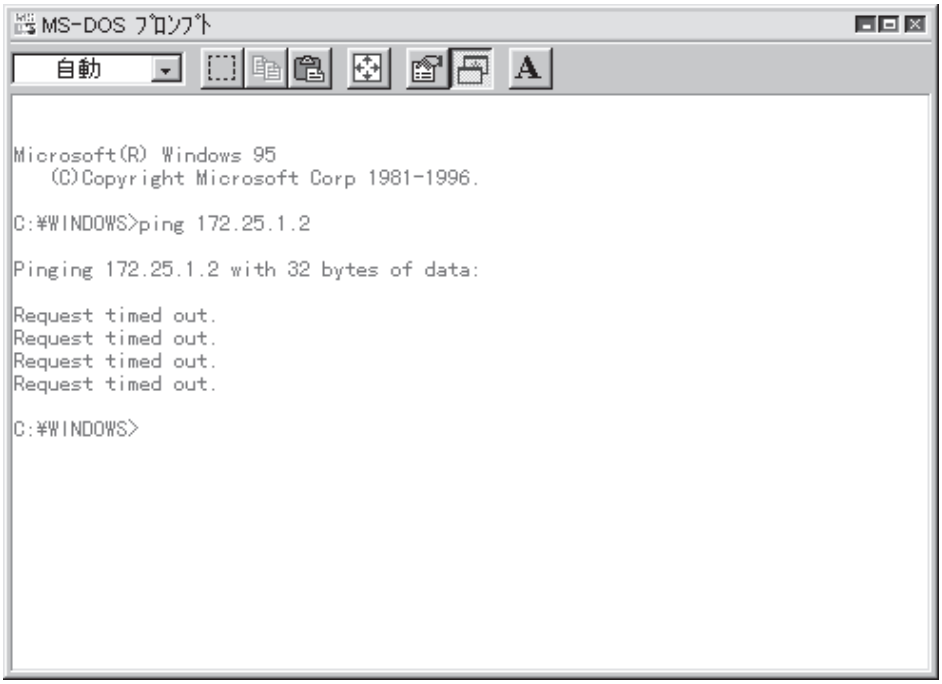
注 : PC に TCP/IP がインストールされていない場合のインストール手順については、Windows™ 95 または Windows NT ユーザーズ・マニュアルを参照してください。

"ping" コマンドを使用して、上記の物理的接続と PC の TCP/IP 構成を確認することができます。"ping" コマンドは、TCP/IP ネットワーク・プロトコルをインストールした Windows™ 95 と Windows™ NT の両方で使用することができます。PC とオシロスコープのネットワーク接続を確認するには、以下の手順に従います。

1. **MS-DOS プロンプトを表示します。**
2. **"ping <ip_address>" と入力します。"<ip_address>" は、オシロスコープに割り当てられたスタティック・アドレスです。次ページのダイアログ・ボックスは成功した "ping" の結果の例で、ここに表示されているイーサネット接続が確立されたことを示します。この場合のオシロスコープの IP アドレスは 172.25.1.2 で、工場で割り当てられたデフォルトのアドレスです。**



この ping コマンドは装置にメッセージを送信し、レスポンスを待ちます。但し、タイムアウトが発生した場合は、次ページのダイアログ・ボックスに示すように、宛先（オシロスコープ）として使用された IP アドレスが不正であるか、PC の IP アドレスのサブネット・マスク内に存在しません。



ネットワーク接続

オシロスコープをネットワークに接続する場合は、その前に、ネットワーク管理者への情報の問い合わせが必要です。ネットワークについて不正なアドレスを指定すると、ネットワークとオシロスコープの両方が予測しない動作をとる場合があります！但し、ネットワーク接続は、ネットワークへのオシロスコープのプラグ・イン同様、単純なものでなければなりません。接続の妥当性の確認は、物理的接続の確認手順に従って行います。

主としてシステム・スループットに関心があるユーザの場合、ネットワーク・トラフィックではオシロスコープ・データの転送速度が遅くなるので、ネットワーク接続はお奨めしません。

注 ： デフォルトのゲートウェイは、“ 172.25.0.1 ”として割り当てられます。このゲートウェイをネットワークが使用していない場合、コンピュータとオシロスコープは同一サブネット上に存在しなければなりません。

IP アドレスの変更

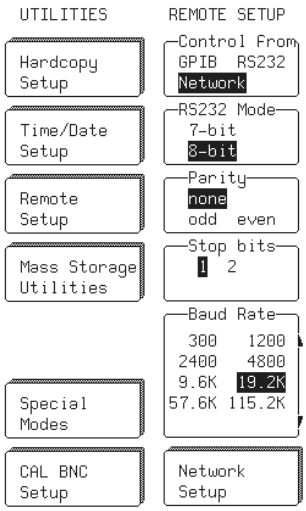
オシロスコープとホスト PC 間の通信の確認が終わったら、Waverunner オシロスコープのフロント・パネル・ユーザ・インタフェースによって、IP アドレスを容易に任意のアドレスに再設定することができます。

IP アドレスを変更したら、オシロスコープ装置は、それ以降、元の IP にレスポンスを送信しません。

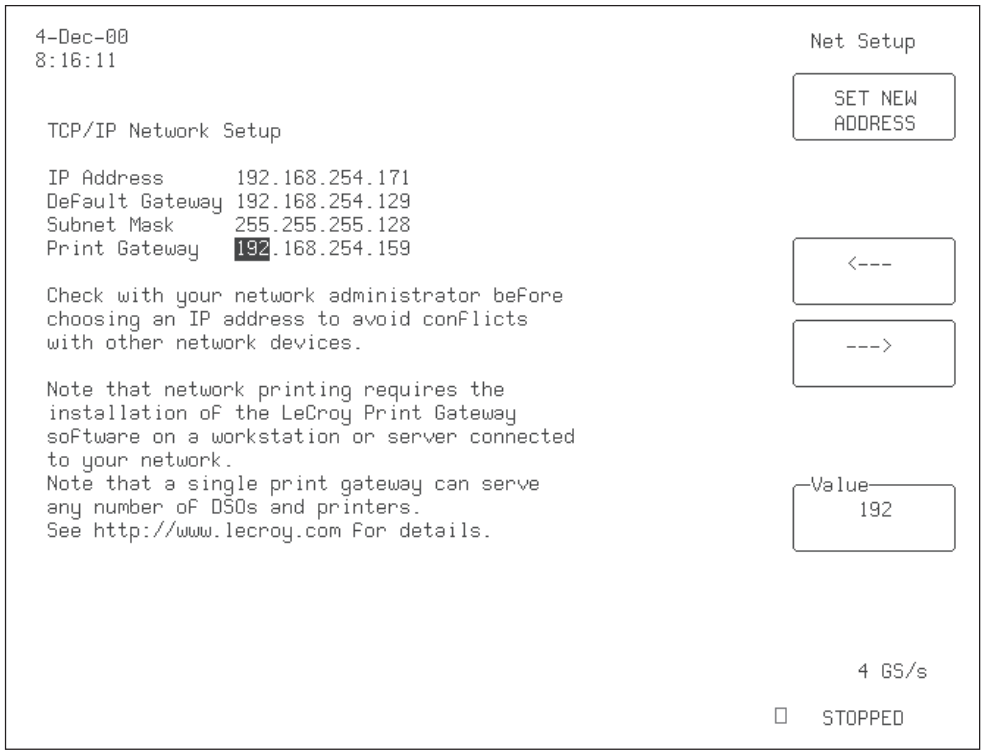
ネットワーク設定が未知であるか、誤って不正な値に設定された場合、以下の手順を実行して工場で割り当てられたデフォルト・ネットワーク設定を復元することができます。

1. オシロスコープ・フロント・パネルの [UTILITIES] ボタンを押します ([Utilities] メニューが表示されます)。

2. フロント・パネル・メニュー・ボタンで、[REMOTE SETUP] を選択します ([REMOTE SETUP] メニューが表示されます)。[Control from] を Network に設定します。



3. フロント・パネル・メニュー・ボタンで、[NETWORK SETUP] を選択します ([TCP/IP Network Setup] が表示されます)。オシロスコープ・フロント・パネル・コントロールによって、イーサネット設定の確認 / 修正を行います。
4. 設定が終わったらメニューボタンで [SET NEW ADDRESS] を選択します。
5. その後、Waverunner の電源を切り、再投入します。


4-Dec-00
8:16:11
Net Setup
SET NEW ADDRESS
TCP/IP Network Setup
IP Address 192.168.254.171
Default Gateway 192.168.254.129
Subnet Mask 255.255.255.128
Print Gateway 192.168.254.159
<---
--->
Check with your network administrator before choosing an IP address to avoid conflicts with other network devices.
Note that network printing requires the installation of the LeCroy Print Gateway software on a workstation or server connected to your network.
Note that a single print gateway can serve any number of DSOs and printers.
See http://www.lecroy.com for details.
Value
192
4 GS/s
STOPPED

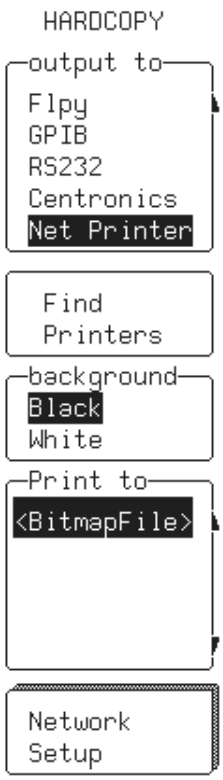
ハード・コピーの作成

ネットワーク・プリンタの使用

☆イーサネットを装着した場合のみネットワークプリンタが使用可能です。

ネットワークに接続した場合、システム対応のネットワーク・プリンタに印刷することができます。この場合、ネットワーク上の任意の Windows Pc に LeCroy DSO Printer Gateway（プリント・ドライバ）をインストールする必要があります。

以下の HARDCOPY メニューを使用して、「NetWORK Setup」メニューにアクセスします。



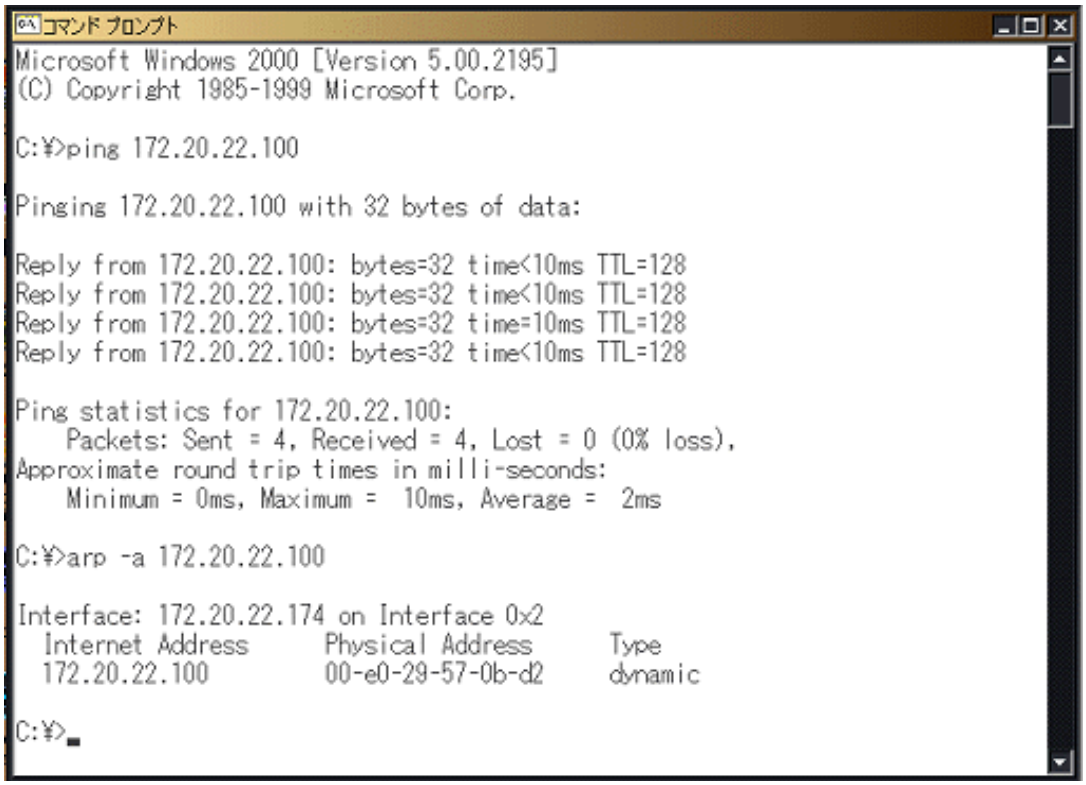
手順

1. 「Net Printer」までスクロール・ダウンします。
2. ネットワーク・アドレスの設定を行う「Network Setup」ボタンを押してメニューを開きます。
3. LeCroy DSO Printer Gateway をインストールしてある Pc の IP アドレスを Print Grteway に設定します。(E-2-6 図参照)

☆ LeCroy DSO Printer Gateway は LeCroy 社のホームページ (http://www.lecroy.com) よりダウンロードできます。

ＭＡＣアドレスの確認方法

ＭＳ－ＤＯＳコマンドプロンプトを実行して
ping （IP address）コマンドを実行した後
ARP -a （IP address） コマンド で確認できます。



はじめに

オシロスコープ装置と別個に提供されるWaverunnerオシロスコープ ソフトウェア・ツールを使用して、ユーザ自身の目的に適したプログラムを迅速かつ容易に開発することができます。このツールには、*ActiveDSO* と *ScopeExplorer* があります。以下に説明するすべてのソフトウェアのファイルは、CD-ROMに入っています。

ActiveDSO

*ActiveDSO*は、MicrosoftのActiveXコントロール技術に基づいて、広く使用されるMicrosoftソフトウェア・ツールの機能を拡張し、Microsoft環境内のプログラミングを容易にします。*ActiveDSO*は、Waverunnerオシロスコープに対するコンピュータのインタフェースと、Visual C++、Visual Basic、または他のすべてのActiveX使用可能アプリケーション内のプログラミングを単純化します。たとえば、Microsoft Excelを使用して、Waverunnerオシロスコープのデータを直接に制御/検索することができます。*ActiveDSO*はターゲット アプリケーションの一部になって、Waverunnerの機能全体に対するシームレスなアクセスを可能にします。

ScopeExplorer

このPCベースの接続ツールは、LeCroy装置と、Windows 95/98またはWindows NT搭載のPCを統合します。イーサネット・ポートによってPCに接続されるWaverunnerオシロスコープは*ScopeExplorer*を使用することにより、データと画像を装置から転送し、コンピュータに保管することができます。さらに、*ScopeExplorer*はLeCroy装置との併用を前提に設計されているため、数少ないキーストロークまたはマウス・クリックだけで、これらの作業を遂行することができます。

ActiveDSOの使用

_ActiveDSO_は、Microsoft環境における高速なプログラム開発に非常に適しています。このActiveDSOプログラムは、MicrosoftによってCOMモデルのサブセットとして開発されたソフトウェア技術ActiveXの1つのコントロールです。

ActiveDSO は、装置とホスト・コンピュータ間の容易なインタフェースを提供することによって、Waverunnerオシロスコープに関するプログラミングを支援します。Visual C++、Visual Basic、またはVisual Basic for Applications（VBA）などのプログラムをこまごましたインタフェースに煩わされずにリモート制御プログラムとして使用することができます。*ActiveDSO*は設計構造で中心的役割を果たし、ActiveXの組込みをサポートする異なる製造業者のソフトウェアの効果的な統合を可能にします。

コントロールのインスタンス化

Visual Basicの関数CreateObjectを使用することによって、ActiveXコンポーネントを複数回インスタンス化することができます。オブジェクトを作成した後、接続メソッドを呼び出すと、オブジェクトが初期化されます。ActiveDSOは、多様なPCデスクトップ アプリケーションからのWaverunnerオシロスコープの制御を可能にします。イーサネットを使用するプログラミングの複雑さが完全にこのコントロールにカプセル化されます。たとえば、Excelマクロ内の10行に満たないVBA（Visual Basic for Applications）で、事前スケールされたWaverunnerオシロスコープの波形データを復元することができます（ActiveDSOのオンラインExcel例を参照）。

ActiveDSOコントロールを使用するには基本的方法が2つあります。

1. OLE Automation互換クライアント（たとえば、Microsoft Word）に埋め込まれた可視オブジェクト（取り込んだ、Waverunnerオシロスコープの表示画面を示す）としての使用。詳細は、以下の「例 : Word」を参照してください。
2. スクリプト言語（たとえば、VBA）によってアクセスする、Waverunnerオシロスコープをリモート制御する不可視オブジェクトとしての使用。詳細は、以下の「例 : VBA」を参照してください。

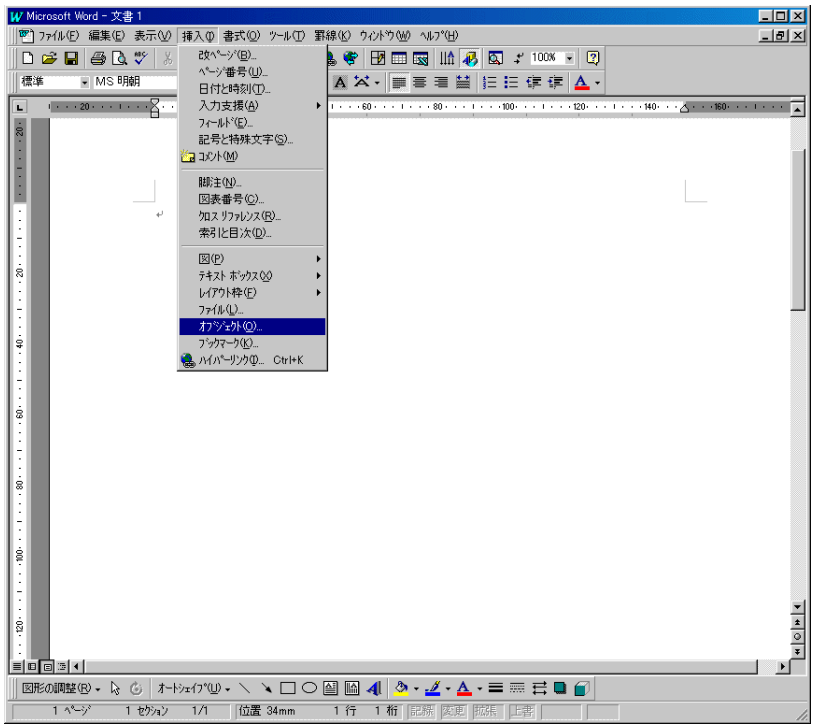
*ActiveDSO*コントロールはActiveX組込み可能クライアントに埋め込んだ後、プログラミングやスクリプト処理を行わずに手操作で使用することができます。

例 ： Word

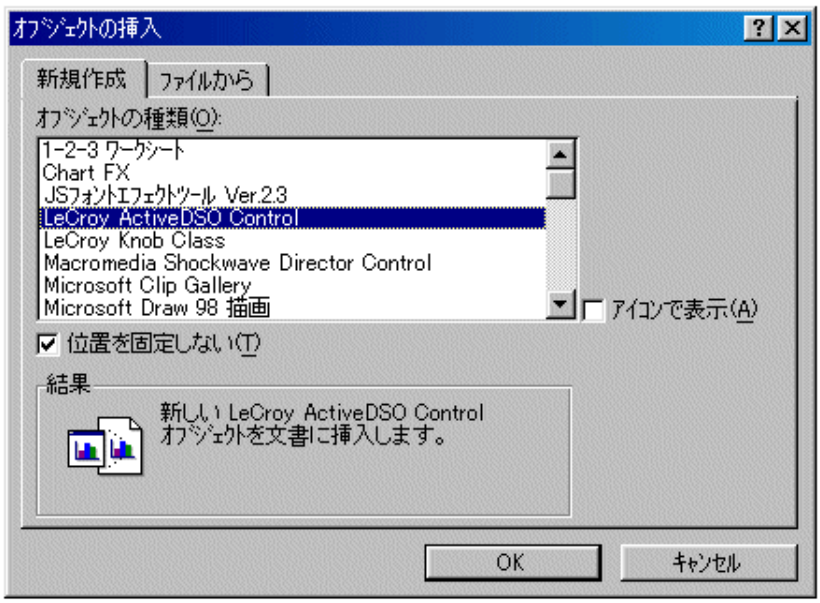
注 ： この例は、Word 97が使用されていることを前提にしています。これより前のバージョンの場合、同一の動作を行う場合と行わない場合があります。

この例は、Microsoft Word スライドに埋め込んだコントロールを示します。数回のマウス・クリックを実行するだけで、Waverunner オシロスコープが取り込んだ波形を Word に簡単にインポートすることができます。

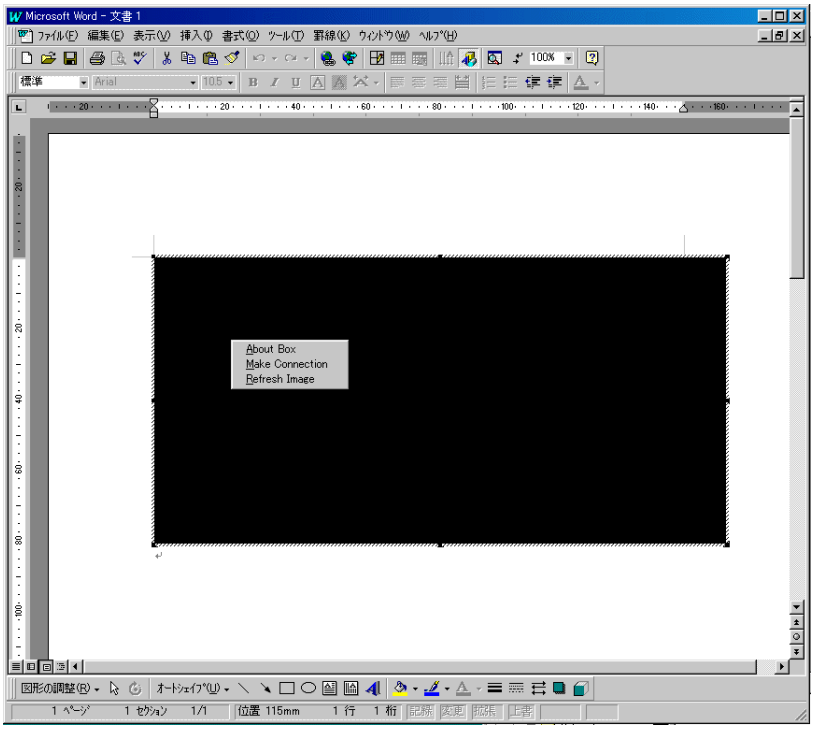
1. CD-ROM の *ActiveDSO* ファイルを PC にインストールします。
2. イーサネット経由で、PC と Waverunner オシロスコープを正しく接続します。
3. Word を新規に開きます。
4. 下の画面に示すように、[挿入] > [オブジェクト] を選択します。



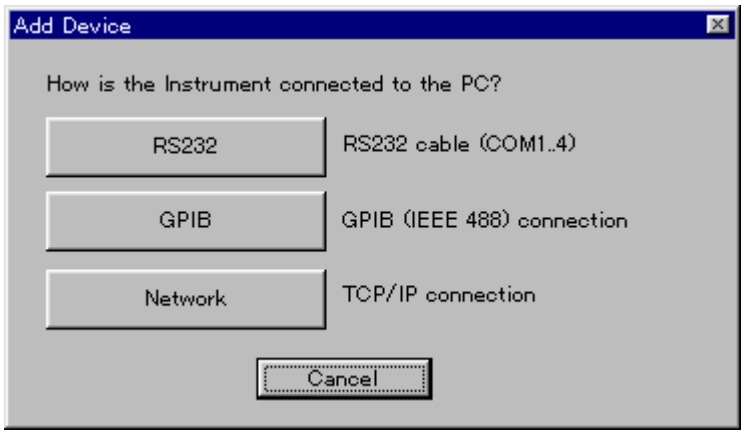
5. 下の画面に示すように、ポップアップ ウィンドウで、[Lecroy ActiveDSO Object]を選択します。



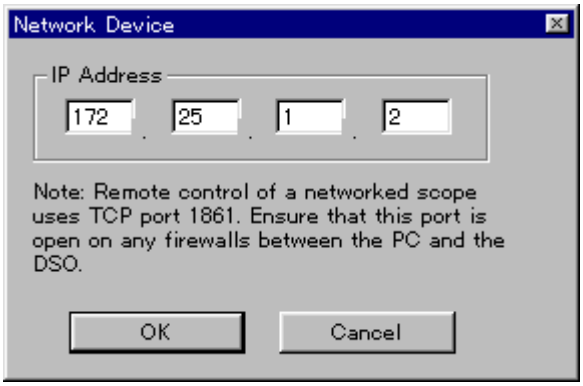
6. オブジェクトを右クリックした後、[Make Connection] を選択します。



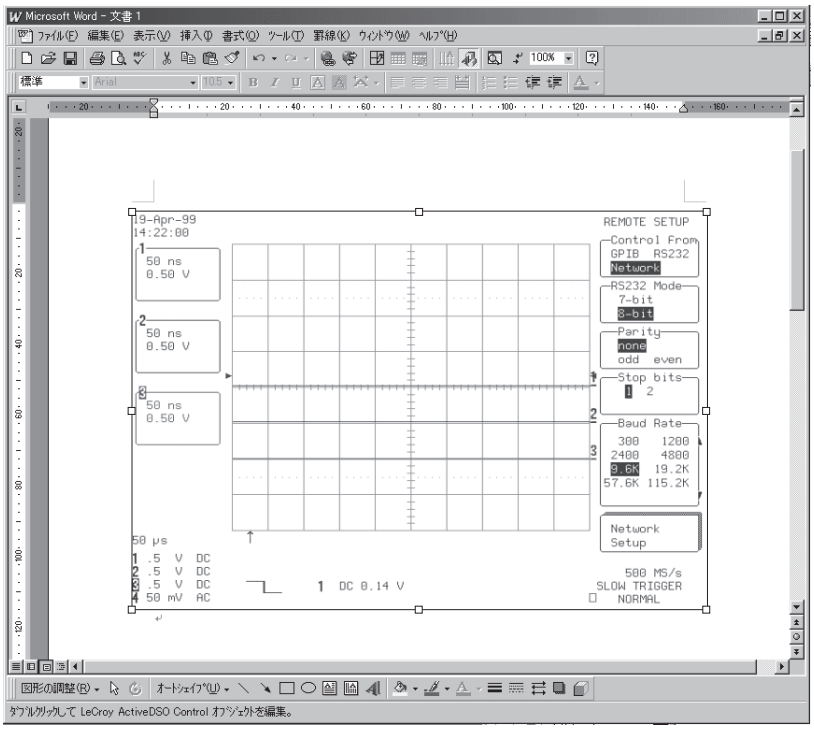
7. 下の画面の [Network TCP/IP connection] を選択します（オシロスコープとして、Waverunner オシロスコープを選択します）。



8. Waverunner オシロスコープの IP アドレスを入力した後、[OK] をクリックします。


Network Device
IP Address
172
25
1
2
Note: Remote control of a networked scope uses TCP port 1861. Ensure that this port is open on any firewalls between the PC and the DSO.
OK
Cancel

9. もう1度、オブジェクトを右クリックして、[Refresh Image] メニュー項目を選択します。下の画面に示すように、取り込んだ波形が表示されます。



Waverunnerオシロスコープで取り込んだ波形がWordにインポートされます。

ActiveDSO™ オブジェクトがアプリケーション内に正しく設定されたら、WriteString ( ) などのオブジェクト・メソッドを使用してマクロ・スクリプトを作成し、"DISP ON, C1:TRA ON, TRMD AUTO" を送信することができます（「リモート・コントロール・マニアル」を参照）。この後、RefreshImage ( ) メソッドを使用して画面を更新することができます。

例 **：VBA**

VBAは、最近の多くのWindowsアプリケーションに組み込まれているプログラミング言語です。それはVisual Basicのサブセットで、OLE Automationサーバおよび ActiveX コントロールの使用を非常に単純なものにします。Waverunnerオシロスコープへの接続とリモート・コマンドの送信が容易であることが、以下のVBAサブルーチンでおわかりになるでしょう。

```
Sub LeCroyDSOTest()
      Dim dso As Object

      Set dso = Createobject("LeCroy.ActiveDSO.1")

      Call dso.AboutBox         Present              the
control's About box
      Call dso.MakeConnection(ÒIP:172.25.1.2Ó)
      Connect to the unit
      Call dso.WriteString("DISP ON", 1)
      Enable the internal display routine
      Call dso.WriteString("TRMD AUTO", 1)     Set
the trigger mode to AUTO
End Sub
```

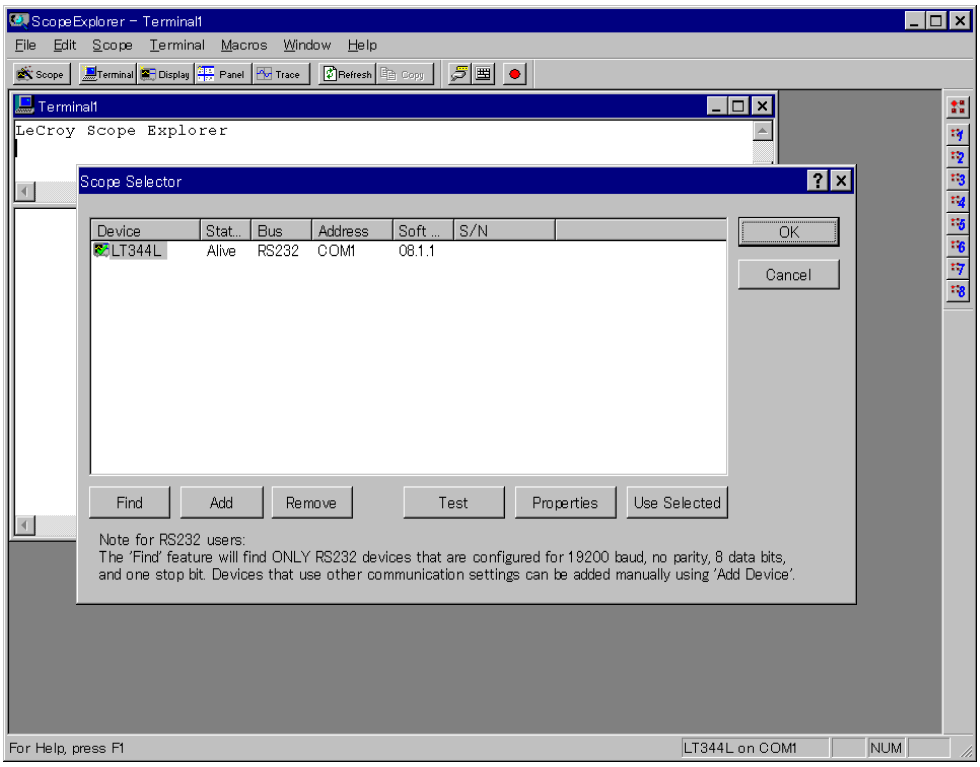
注　：　詳細は、ActiveDSO オンライン・ヘルプを参照してください。オンライン・ヘルプには、ActiveDSO メソッドおよびプロパティを示す VisualC++ の例が含まれています。

Microsoft Officeソフトウェア群のメンバ内でVBAエディタを使用する場合、以下の手順に従います。

1. **[Tools] > [Macro] > [Visual Basic Editor] メニュー項目を選択します。**
2. **VBA ウィンドウが表示されたら、[Insert] > [Module] を選択します。**
3. **上記の VBA 例を表示されたエディタ・ウィンドウにコピーします。**
4. **テキスト・カーソルをサブルーチン内に位置づけます。**
5. **[Run] > [Run Sub/UserForm] を選択するか、F5 キーを押します。**

*ScopeExplorer*の使用

1. *ScopeExplorer*を起動します。
2. [Scope] > [Scope Finder] を選択します。
3. 下の画面に示すように、[Scope Selector] ウィンドウで、[Add] をクリックします。
4. [Add Device] ウィンドウが表示されたら、[Network] を選択します ([Network] ボタンが表示されない場合は、ALT + N を押します)。
5. E-3-8 ページの図に示すように、[Network Device] ウィンドウが表示されます。
6. [Network Device] ウィンドウで、Waverunner オシロスコープの IP アドレスを入力します。


ScopeExplorer – Terminal1
File Edit Scope Terminal Macros Window Help
Terminal1
LeCroy Scope Explorer
Scope Selector
Device Stat... Bus Address Soft ... S/N
LT344L Alive RS232 COM1 08.1.1
OK
Cancel
Find Add Remove Test Properties Use Selected
Note for RS232 users:
The 'Find' feature will find ONLY RS232 devices that are configured for 19200 baud, no parity, 8 data bits, and one stop bit. Devices that use other communication settings can be added manually using 'Add Device'.
For Help, press F1
LT344L on COM1
NUM

ScopeExplorer の機能

- ***ターミナル機能 ： この機能により、リモート制御コマンドを送信して、データを検索することができます。***
- ***画面の取り込み ： [Display] ボタンを使用して、取り込んだ波形の、オシロスコープ内部 "スクリーン" を表示することができます。[Refresh] ボタンを使用して、"スクリーン" ダンプ画面をリフレッシュします。***
- ***IP アドレスの変更 ： ScopeExplorer を使用して、Warunner オシロスコープの IP アドレスを変更することができます。詳細は、第 2 章を参照してください。***

ScopeExplorer はすべての LeCroy 装置でサポートされており、詳細情報は LeCroy の Web サイト（www.lecroy.com）にあります。

製　品　保　証

この製品は、お客様に安心してお使い頂くために下記の保証をいたします。

◆ **保 証 期 間**　ご納入後１年間保証いたします。

◆ **保 証 条 件**　万一、保証期間内に当社の責任による不測の故障などが生じた場合には無償修復いたします。
連絡先は、別紙のサービスネットワークをご利用ください。

■ お問合わせ窓口

岩 通 計 測 株 式 会 社
岩通ティー・エム・イー・サービス株式会社

		ハローイワツウ
技術的な、取扱い・測定方法など	**計測フリーダイヤル**	**0120-086-102**
修理納期など	**会津サービス課フリーダイヤル**	**0120-267-905**

● URL : http://www.iti.iwatsu.co.jp　● E-mail : info-tme@iwatsu.co.jp

お願い：セールスネットワーク、サービスネットワークの最新情報は、当社のホームページまたはフリーダイヤルでご確認いただくようお願い申し上げます。